July 2019

An Empirical Analysis of Network Traffic: Device Profiling and Classification

Mythili Vishalini Anbazhagan

AN EMPIRICAL ANALYSIS OF NETWORK TRAFFIC: DEVICE PROFILING AND CLASSIFICATION

A Thesis Presented

by

MYTHILI VISHALINI ANBAZHAGAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE

May 2019

Department of Electrical and Computer Engineering

AN EMPIRICAL ANALYSIS OF NETWORK TRAFFIC: DEVICE PROFILING AND CLASSIFICATION

A Thesis Presented

by

MYTHILI VISHALINI ANBAZHAGAN

Approved as to style and content by:

David Irwin, Chair

Daniel Holcomb, Member

Michael Zink, Member

Robert Jackson, Department Head
Department of Electrical and Computer Engineering

ABSTRACT

AN EMPIRICAL ANALYSIS OF NETWORK TRAFFIC: DEVICE PROFILING AND CLASSIFICATION

MAY 2019

MYTHILI VISHALINI ANBAZHAGAN

B.E., KUMARAGURU COLLEGE OF TECHNOLOGY

MSECE, UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor David Irwin

Time and again we have seen the Internet grow and evolve at an unprecedented scale. The number of online users in 1995 was 40 million but in 2020, number of online devices are predicted to reach 50 billion, which would be 7 times the human population on earth. Up until now, the revolution was in the digital world. But now, the revolution is happening in the physical world that we live in; IoT devices are employed in all sorts of environments like domestic houses, hospitals, industrial spaces, nuclear plants etc., Since they are employed in a lot of mission-critical or even life-critical environments, their security and reliability are of paramount importance because compromising them can lead to grave consequences.

IoT devices are, by nature, different from conventional Internet connected devices like laptops, smart phones etc., They have small memory, limited storage, low processing power etc., They also operate with little to no human intervention. Hence it becomes very important to understand IoT devices better. How do they behave

in a network? How different are they from traditional Internet connected devices? Can they be identified from their network traffic? Is it possible for anyone to identify them just by looking at the network data that leaks outside the network, without even joining the network? That is the aim of this thesis. To the best of our knowledge, no study has collected data from outside the network, without joining the network, with the intention of finding out if IoT devices can be identified from this data. We also identify parameters that classify IoT and non-IoT devices. Then we do manual grouping of similar devices and then do the grouping automatically, using clustering algorithms. This will help in grouping devices of similar nature and create a profile for each kind of device.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The Internet as we know it, keeps changing rapidly. When the Internet was first brought to the general public, it only had computers connected to it through some form of wired connection. Then came the next phase of the Internet with the advent of WiFi. This led to the rapid increase in the number of mobile devices and laptops that became connected to the Internet. As time progressed, the number of devices connected to the Internet went from 500 million in 2003 to 12.5 billion in 2010, a 2400% increase [1]. Cisco believes that the IoT (Internet-of-Things) wave began in this period, when the number of devices connected to the Internet exceeded the world population. And it is predicted that the number of Internet-connected devices will reach 50 billion by 2020.

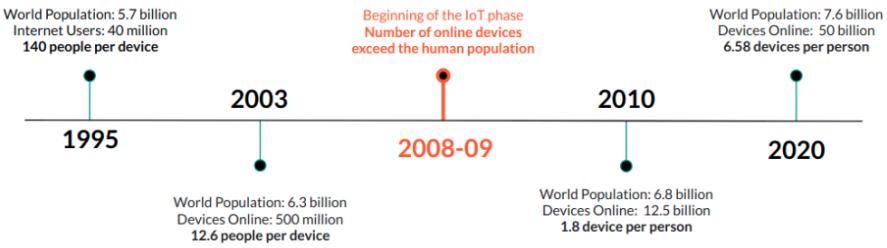

Figure 1.1: The Beginning of the IoT Phase

The concept of IoT was conceived long ago in the late 90s [2]. But we see an explosion in the number of IoT devices only now. Because, back then, we did not have the technology to realize this concept cost-effectively. But today,

- manufacturing of electronic devices has become faster and cheaper

- more and more devices are getting wireless communication capability
- network access has become better with high speed broadband connections
- battery technology has become more efficient
- we are moving into IPv6 to allow a larger number of devices to join the Internet

Leveraging these technological improvements, things from all walks of life have joined the realm of IoT; for instance, light bulbs, thermostats, door bells, windows in houses, air purifiers and humidifiers, garage doors, refrigerators, dishwashers, washers and dryers etc., in the domestic sector; Bluetooth beacons to track the location of assets, tracking devices in oil and gas facilities to detect oil spills, sensors in cargo shipments to track the condition of the environment like temperature, humidity, tilt, pressure etc., in the industry sector; Fitbits for tracking heartbeat rate or number of steps taken in a day, insulin pumps with sensors for monitored and controlled release of Insulin and so many more devices in the medical sector.

Hence, an IoT device can be defined as, "*A device whose function focuses on sensing and/or actuating on the environment around them*".

Up until now, the revolution was in the digital world. But now, the revolution is happening in the physical world that we live in; IoT devices are employed in all sorts of environment like domestic homes, hospitals, industrial sectors, nuclear plants etc.,

CHAPTER 2

MOTIVATION

In order to perform the functions they are intended to perform, IoT devices must collect data about/from their environments; they collect data 24 hours a day, 7 days a week. And they process that data (or more correctly, a remote server does the processing for them) to derive meaningful insights about their environment and make intelligent decisions. Further, IoT devices are designed in such a way that they require little to no human intervention for performing their duties. As a result, if they are compromised, it is very difficult for a consumer to realize that they have been compromised. And IoT devices are typically endowed with very low system resources, like low processing power, small memory and storage etc., Because of this, their computation power is lower compared to traditional Internet connected devices and it is difficult to implement the same security mechanisms that we do in devices like laptops or smart phones. As a result of all the above factors combined, the privacy and security risks posed by IoT devices are novel. And because they are employed in a lot of mission critical and safety critical applications, their compromise can have grave or even fatal consequences.

The huge amount of risks involved is the main motivation for this thesis. The lack of understanding about the IoT devices lets us make uninformed decisions on buying and using the IoT devices. Hence, this thesis is aimed at understanding the IoT devices better; that would be the first step in defining better security policies for handling IoT devices and safeguarding ourselves from giving away sensitive information in the name of data collection.

2.1 Privacy Risks in IoT

In smart homes, IoT devices collect lots of information that could be used to discern information that were not intended by the occupants of the house. For example, data from smart light bulbs, that use motion detectors to turn on the lights, could be used to find out information about the current occupancy of the house. Or if the ISP sees that medical IoT devices are connected to the Internet from a particular house, they could sell that information to advertising agencies which can direct health care advertisements towards the residents of that house. Revealing that they own a Blood Pressure Monitor is, by itself, a breach on privacy; the ISP does not have to know the exact readings on the Blood Pressure Monitor.

And the most alarming aspect of this data collection is that it is very difficult to understand what is being collected, when it is collected etc., and to verify if a device collects only the data it says it will collect.

For example in 2015, there was an issue in Chromium, which is the open source web browser from Google. Some users noticed that, when they installed Chromium, an extension namely "Hotword" was downloaded and loaded automatically into the browser. There was no way to either stop this extension from being downloaded or disable it nor was it possible to view the source code to see what it was doing. Not being able to view the source code is itself a violation of the Open Source Policy, which Chromium has adopted. Google answered that it was to activate the microphone, so that it could catch the keywords "OK Google", to prompt Google Assistant. This could potentially mean that the microphone was ON the entire time and it was recording everything that was happening around it, in order to detect the keywords "OK Google." Google went on to explain that being enabled does not mean that the microphone was always turned ON, but only when the tab opens "Google.com" or when a New Tab with Google Search Engine is opened. But after severe revolt from its users, that extension was removed from Chromium.

Also there is the risk of Identification i.e., taking the data collected like location, address etc., and associating that data with an individual. This could potentially lead to profiling and tracking of that person, recording every move they make, recording wherever they go, without their consent [3]. The increasing number of voice activated devices are also adding to the privacy risks of identification and tracking. As every person's voice is so different, to the point of unique identification based on pitch, gaps between words, how they pronounce the words etc., voice can be used effectively to identify a person among crowd [4]. Many smart phones use Machine Learning to learn and recognize their owner's voice and do not respond to anybody else's voice. With surveillance cameras cropping up every where on the grounds of offering security, the risk is ever increasing. It is possible to track all the places someone has been for an entire day in any major city, where there are traffic cameras on almost all the streets.

Hence, with the growing number of IoT devices in every house, it is very difficult for a consumer to understand what data is being collected about them and control it. IoT devices often come with a manual that state the purpose of the device but its content is seldom in simple, understandable words. And the manufacturers and sellers of the IoT devices also rarely take the pain to remain transparent about the exhaustive list of data that is being collected by their devices. With more and more shops and cities opting to use these IoT devices for various purposes, the general public's chances of staying away from these device is also growing slim. Governments have still not caught up on the recent trends and technologies and their regulations are not very effective in preventing and curbing these privacy breaches.

2.2 Security Risks in IoT

IoT devices can be compared to babies with very weak immune systems in a world of adults with stronger immune systems who can fight off infections much more easily than the babies do. IoT devices, by design, are small systems and are meant for

doing one or a very few closely related tasks. They also operate on battery power. Hence their system resources are not as powerful as mainstream computers or smart phones. Hence, by default, their software security mechanisms are weaker; at present, they cannot fend for themselves as powerfully as other Internet-connected devices do. Also, they do not have enough hardware resources to properly encrypt the data they collect and then send them out; this encryption and other heavy lifting are mostly done by IoT Gateways. Further, for most of these devices, their default passwords are not changed at all after they are bought. This is a critical flaw, one that invites hacking.

There has been a lot of security breaches and threats recently, after the influx of large number of IoT devices into the Internet. Taking advantage of the above-mentioned short comings, IoT devices have been victims of a lot of hacking attempts. The best example is the October 2016 DDoS (Distributed Denial-of-Service) attacks targeted towards Dyn, a major DNS service provider [5]. In a DoS attack, a huge amount of traffic/requests is aimed towards servers so that they will either become unavailable to legitimate requests while busily handling this traffic surge and not completing any request or will be overwhelmed completely and shut down. It is 'distributed' in the sense that the request comes from hosts distributed all over the Internet. The attack on Dyn was launched using a botnet, a network created by infecting devices with a malware (malicious software) and then controlling them without the owner's knowledge. This attack used a malware called Mirai, that infected devices that had not changed their default passwords and took control of them. Further, these infected devices started searching the Internet for more and more devices that were using default passwords and took them over too. It brought down major websites like Twitter, PayPal, Etsy, Verizon, Comcast etc., Not changing the factory-provided default password is a huge security risk because the default passwords are often iden-

tical for a single line of products from the same vendor. It is very easy to find them online in product documentation or other sources [24].

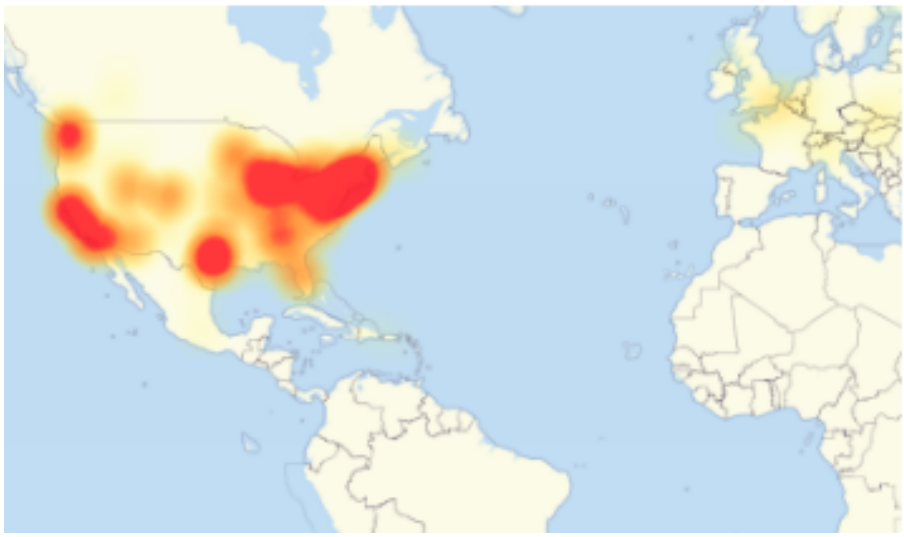

Figure 2.1: Internet Outage Caused by DDoS Attack on Dyn - Oct 21, 2016

Another well known DDoS attack is the one on KrebsonSecurity. Though it was unsuccessful, the scale of the attack was unprecedented [6]. The traffic targeted towards their servers were close to 620 Gigabits per second. The security experts from Akamai dealt with the attack and curbed it successfully after much effort, but said that it was double the scale of the largest attack they had ever seen before that. Infecting devices with malware and conducting DDoS attacks have happened before too. And many times, DNS servers are the target; because they are mostly public and accept requests from all addresses. Instead of targeting single servers, targeting DNS servers and rendering them useless will affect countless servers and can even bring down a part of the Internet. Before IoT devices, DNS Amplification attacks were often used. In this attack, the requests are crafted in such a way that they will elicit a very large response from the server. Hence, with a fewer number of requests, it will be possible to bring down the DNS servers. Now, with the number of IoT devices connected to the Internet approaching billions, even without amplification

DNS attacks, it is still possible to flood DNS servers by utilizing the sheer number of IoT devices. If these devices, already having strength in numbers, were to wage amplification DNS attacks, the magnitude of that attack can bring down the entire Internet one day.

DDoS attacks are just one of the security risks surrounding the IoT devices. There are many more like Man-in-the-middle attacks, Sinkhole attacks, Sleep Deprivation attacks etc., All are made easily possible by taking advantage of the flimsy security patches on IoT devices.

IoT devices have also invaded the industrial space recently. This is said to be the fourth industrial revolution [7] after mechanical, electrical and IT revolution. Security breaches in the industrial sector can be even more grave and sometimes even fatal. A mishap in a sensor that detects oil leaks could lead to very serious environmental impacts and can cost millions of dollars to rectify it. Failing of or malware infection in power grids could lead to major outages. In 2010, nuclear facilities in Iran were damaged by Stuxnet, a malicious worm that specifically targeted Siemens SCADA (Supervisory Control and Data Acquisition) systems and infected the Programmable Logic Controllers (PLCs) [29]. This compromised the valves in the gas centrifuges and as a result, pressure built up inside the centrifuges ultimately damaging them. In 2015, security researchers Charlie Miller and Chris Valasek of Cruise Automation demonstrated that they could hack into a 2014 Jeep Cherokee and take control of the music player, air conditioning system and even cut down the transmission [8]. Hence, compromising of Industrial IoT devices are serious issues that require more attention and consideration.

As more and more devices enter the Internet each day, large in numbers and varying in types, the privacy and security risks involved keep increasing. New methods of hacking these devices crop up each day with the advancement in technologies and tools. Therefore the first step in preventing these attacks and mitigating these risks

is to understand the IoT devices better. That is the aim of this thesis. It is focused on learning more about the behaviour of the IoT devices such as what type of data do they collect? what kind of information can be discerned from the collected data? how different are they from traditional Internet connected devices? do IoT devices behave the same way in the network as other devices do? can IoT devices be identified just from their network traffic? more importantly, is it possible to identify them just from the network data that leaks outside, without even joining the network? If an intruder is looking to compromise a network, he would be looking for the weakest link in the network and that would be IoT devices. If can identify IoT devices just from their network data that leaks outside, without even joining the network, then the network is as good as compromised.

CHAPTER 3

RELATED WORK

IoT data analysis is a rapidly growing field of research today in both academy and industry. IoT devices are relatively new to the world of computer networks; hence more and more research is under taken everyday to better understand them. A cloud based architecture for IoT devices is proposed in [9] for fulfilling the need of a scalable and secure IoT cloud network which would be essential for future projects like smart cities. The research in [10] tries to bridge the gap between IoT data analysis and machine learning by proposing a framework to increase the ease of development. Another framework for IoT is proposed in [11] to improve edge computing by incorporating a controller in the fog. Many researches also focus on improving the security in IoT networks. Authors from [12] review the state of security mechanisms in IoT and the status of the research that is going on in this field regarding encryption algorithms, communication security, protecting the sensor data etc., Finally they also list the possible challenges in IoT in incorporating the security mechanisms. The work in [13] tries to list out the environmental and security context and cross-device interactions of the IoT devices through "brute-force" and proposes using such models for identifying potential attacks on the IoT devices in the network. Though many such innovative attempts are undertaken more and more, research on analyzing the nature of the IoT devices itself is very few. Few researches so far have attempted to understand the IoT devices through their network traffic data. The work done in [14] provides a list of various techniques that can be used for classifying the network traffic using machine learning. It highlights the limitations of conventional packet

inspection and classification purely based on port numbers and payload. The research in [15] aims to detect malware in the network by using supervised learning to analyze the network traffic and detect the presence of malware and trying to attribute it to an already known group of malwares. This is in a way characterizing the network traffic but does not use this approach to study IoT devices directly. The work done in [16] also tries to detect malware by characterizing the network traffic in a smart home environment. To evaluate their model, they conduct attacks on real IoT devices and succeed in accessing an IoT device from outside the network. Research done in [17] is more closely related to our research. It aims to identify the IoT devices using their network characteristics and employing supervised learning to further train and test their model. But the number of devices considered are very few (around 13) and network characteristics considered are primarily TCP sessions. Some publicly available data from Alexa and GeoIP were used to enrich their data. Though it shows a high accuracy of results, the data set considered is very less. A hybrid approach based on Convolutional and Recurrent neural networks to classify the network traffic is proposed in [18] as an alternative method for classifying traffic over using supervised or unsupervised learning. Authors from [17] also carried out another work [19] where they aim to detect unauthorized IoT devices by using the supervised machine learning algorithm Random Forest. But here also, the number of IoT devices considered for the experiment is 17. The work done in [20] also aims to characterize and profile IoT devices using network data. They have a test bed comprised of 21 unique IoT devices and try to simulate a smart city environment. More device characteristics like their Sleep/Active periods and the application layer protocols of the devices are considered. And finally they use supervised learning to identify devices. This research also shows high accuracy of results but the period of the network data capture is 3 weeks, of which 2 weeks is used for analyzing and

training and 1 week of data is used for testing. It is also carried out in a simulated lab and data is captured only from inside the network.

As per the best of our knowledge, up until this point, a full scale real time analysis of the IoT devices has not been carried out to learn the characteristics of the devices itself. Large enterprises do employ IoT network data analytics but they are aimed at assisting their business, helping them in decision making, in operations management etc., Some academic researchers also have attempted to characterize the IoT devices using their network traffic data but the experimentations carried out were often small with a maximum of 20+ devices being employed. The devices were hand chosen and were usually employed in a lab setup. Data was generally collected only from inside the network. The duration of the data capture was also short, in the order of weeks. No study has collected data for comparatively longer duration and from various points in the network in a real-time setup to purely learn about the IoT devices. That is what this research is trying to fulfill. It captures network data from 40+ IoT and non-IoT devices in a real-time environment. The minimum period of capture in our experiment is 3 months and the maximum is 6 months for analysis and testing. We have captured data from both inside and outside the network to learn what can be learned about the devices in the network just from their network traffic, observed from both outside and inside the network. Also a lot of the recent researches that have used machine learning for characterizing the network traffic have used supervised learning. But we are aiming to apply clustering algorithms to group devices that are innately similar in nature. The profiles thus obtained would help in identifying IoT devices and non-IoT devices in the future and possibly go a step further and identify the device itself.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Capturing the Network Data

The prerequisite for our analysis is the network data. The network data can be defined as what is sent across the network between the nodes present in the network. By collecting this data from various points and analyzing it, we can find out what information can be learnt from it. To capture this network data, we used Tshark [21], which is the command line version of the popular tool, Wireshark [22]. Tshark is used for capturing, displaying and parsing the network data into human readable format. The network data is sent across the network as binary data in the form of packets and the captured data is stored with the extension '.pcap', which stands for 'packet capture'. Cron jobs were used for ensuring the continuous running of Tshark to capture packets without a break. Since the we are going to capture the network packets at a remote location, the entire setup is implemented onto Raspberry Pi devices and deployed at the target location. Since the Raspberry Pi has limited storage, the collected data is moved periodically onto a remote server for storage.

4.2 Points of Traffic Observation

From the network's point of view, the traffic can be observed from either inside the network or outside the network. An intruder might not know the password to join the network, but he might still observe the traffic from outside the network, capturing all the packets that he can. Sometimes, an attacker might gain access into the network and in that case, he can probably see almost everything that is going on inside the

network. To understand what information can be learned just by passively observing the network traffic without joining the network and what can be learned once access is gained to the inside of the network, we are going to capture the traffic from both points - inside and outside the network.

4.3 Modes of Operation of the NIC

In network communication, each device sends and receives data via a hardware called Network Interface Controller(NIC). NICs are designed such that, once they receive a packet, they check if it is meant for their host device. This is done by checking if the destination address in the packet is the same as its own. If so, the packet is accepted and passed onto to the host device. If not, the packet is dropped.

Since we want to capture the packets from all the devices in the network for our analysis, we must make the NIC capture all the packets, regardless of the destination address. This is enabled by changing the mode of operation of the NIC. The following are the different modes of operation of the NIC.

- **Normal Mode** When operating normally, the NIC is associated with a network and drops the packets unless the destination address is specifically its own, or it is a broadcast or multicast packet.
- **Promiscuous Mode** When the NIC is set to operate in Promiscuous mode, it is still associated with a network, but it no longer filters packets that are meant only for itself. It takes all the packets that it receives and passes it up to the CPU. Since it is associated with a network, it cannot catch packets that do not belong to the network that it is associated with.
- **Monitor Mode** An interface, when set to operate in monitor mode, becomes disassociated from any network. Hence, it catches all the packets that are in the air within its reception range, regardless of the network the packets belong to. Only wireless interfaces can be put in monitor mode.

Hence to capture the network traffic externally, without joining the network, we will use a Raspberry Pi device with one of its wireless interfaces set to monitor mode. To capture the traffic from inside the network, we would use another Raspberry Pi device with one of its wireless interfaces set to Promiscuous mode. This will ensure data capture from both inside and outside the network. A single Raspberry Pi device with two interfaces operating in promiscuous and monitor mode respectively can also be used. But for the sake of efficiency and to prevent writing to the full capacity of the internal storage of the Raspberry Pi, two separate devices are used.

4.4 Places and Number of Traffic Captures

The traffic capture was set up in two places. Both are real-time home environments; one has a lot of IoT devices and the other one does not. Lets call the former Location A and the latter Location B.

- **Location A** At Location A, there are two networks - one is protected and the other one is not password protected. The traffic is observed in two different captures, one in Promiscuous mode within the protected network to catch the internal traffic and the other in Monitor mode to capture all the traffic within the range and hence will capture the entire traffic from the unprotected network too. Location A is an individual house with nearby houses present a little distant from each other. The duration of the data capture from outside the network and on internal wired is 6 months, of which 4 months of data is used for analysis and profiling and the rest of the data is used for testing our analysis. The duration of the data capture on internal wireless is 3 months, of which 2 months of is used for analysis and one month of data for testing the analysis.
- **Location B** At Location B, there is only one network, which is protected. The traffic is observed in two different captures, one in Promiscuous mode to capture the internal traffic and the other in Monitor mode to capture all the

traffic within the reception range. Location B is an apartment in an apartment complex setting with multiple apartments present very close to each other. The duration of the experiment is 3 weeks.

Location A is the one with a lot of IoT devices and is the primary target for our analysis. Location B does not have a lot of IoT devices. It is done to verify the integrity of the primary analysis - to test that what is done in Location A can be repeated else where too.

CHAPTER 5

ANALYSIS OF DATA CAPTURED FROM OUTSIDE THE NETWORK

Earlier we discussed that an intruder might just observe the traffic from outside the network, without even joining the network. This poses a very serious security risk. Because, we can prevent intruders from joining the network, but cannot stop them from passively observing the traffic from outside. By capturing the packets that come within their range, they might be able to learn a lot about our network, without our consent. Hence to learn what information can be learned just from the packets captured from outside the network, we will start this analysis.

Before delving deeper into the analysis, we should know what information is visible from this point.

5.1 Data Encapsulation

Encapsulation is the wrapping of information, namely the headers, around the actual data, namely the payload. This wrapped information contains the details necessary to help facilitate the proper delivery of the packet to its intended destination.

The encapsulation varies depending on whether the data is transmitted over a wired medium or a wireless medium. Since we are observing the data from afar, without joining the network, we are going to capture wireless packets at this point. A wireless network is composed of clients and one or more Access Points. Clients communicate with each other and with the outside network via the Access Point. This is called an Infrastructure network. There is another type of WLAN called Ad-hoc, wherein clients directly communicate with each other without an Access Point.

This is not very scalable and hence Infrastructure networks are commonly deployed. In this setting, a wireless packet is encapsulated using the WLAN(Wireless LAN) header.

Table 5.1 on page 18 shows a subset of the various fields that are present in an IEEE 802.11 (IEEE specification for WLAN) header. This header is visible when observing the packets passively even when the network is password-protected. The rest of the contents of the packets - including the headers of OSI layers, are encrypted and not visible at this point if the network is password-protected. With these visible fields and the information available in them, we would try to get some insights about the various devices present in our network. This would be possible for anyone passively observing the traffic too, even without joining the network.

Table 5.1: A subset of fields present in the IEEE 802.11 Header

Field Name	Description
frame.time	Timestamp of the packet capture
frame.interface_name	The interface on which the packet was captured
frame.length	Packet Length
frame.time_delta	Time elapsed after the capture of the previous packet
frame.protocols	Protocols present in the packet
radiotap.dbm_signal	Signal Strength (SSI Signal)
wlan.fc.type	802.11 Packet Type (Data, Management, Control)
wlan.fc.subtype	802.11 Packet Subtype
wlan.fc.fromds	Set to 1 if the packet is coming from DS
wlan.fc.tods	Set to 1 if the packet is going towards the DS
wlan.fc.pwrmgt	To indicate if clients are in Active or Sleep Mode
wlan.ra	Receiver Address
wlan.ta	Transmitter Address
wlan.da	Destination Address
wlan.sa	Source Address
wlan.bssid	BSSID or AP Address
wlan.ssid	SSID or Wi-Fi network name

5.2 Identifying the various Networks present in the vicinity and their degree of proximity

There are 3 types of frames in 802.11 namely - Data, Management and Control Frames.

- **Data:** Data frames are used to carry the actual data from the upper layers.
- **Control:** Control frames are used to deliver the data by handling inherent issues in wireless networks like collision avoidance and Hidden Node problems.
- **Management:** Management frames are used to maintain the wireless network. Devices join and leave the network, get authenticated etc., all using Management frames.

Table 5.2: Location A - List of Networks

Network	Signal Strength (dBm)
KT	-91
162	-85
165	-85
166	-85
KelleyTransit-Guest	-87
TC8717TA5	-89
Verizon-MiFi7730L-1678	-91
HostNetwork	-21
HostNetworkGuest	-22
HP-Setup 14-M277	-43
DIRECT-roku-981-241500	-62
flynn_Ext	-89
Vehicle Hotspot	-88
WIFIFC3D5E	-90
WiFi Hotspot 0306	-89

Beacon frames are a subtype of Management frames that are transmitted periodically by Access Points in a network. They contain the field wlan.ssid which is the network name. The field Signal Strength in Beacon frames indicates the proximity and is given by radiotap.dbm_antsignal. Beacon frames can be identified by the type_subtype value under Frame Control fields. This value is given by wlan.fc.type_subtype and for Beacon frames, its value is 8.

Hence by extracting the signal strength information from Beacon frames (for which the source is the Access Point), we can more or less guess the degree of proximity of the Access Point from our point of observation of the traffic and hence, come to a rough estimate of the network's degree of proximity. The signal strength is represented in Decibel-milliwatt (dBm) in logarithmic scale. The closer the number is to zero, the higher the value; that means the signal strength is stronger and hence, the network is nearer. A number far from zero indicates a weak signal and a distant network.

- **Location A** We were able to identify 15+ different SSID's from the captured data. Table 5.2 on page 19 lists the networks identified at location A. The signal strength mentioned is an average over the signal strength collected from many days of capture data.

  And this data correlates with the fact that 'HostNetwork' and 'HostNetworkGuest' are the home networks at Location A as we can see that their signal is the strongest (signal strengths are 21 dBm and 22 dBm respectively). We also see that the number of rest of the networks in proximity are few and most of them are quite far from our point of observation (abrupt decrease in the signal strength).

- Location B

  Table 5.3 on page 22 shows the list of networks collected from Location B. Nearly 35+ networks were identified. The signal strength mentioned is an average over the signal strength collected from many days of capture data.

  'NeruppuDa' is the name of the host network from where we observed the data and we can see that it has the highest signal strength among all networks (-35.04 dBm) indicating the closest proximity. Also the number of networks within the observation range and the gradual decrease in signal strength shows that this location is an apartment setting where there are more number of devices per

unit area compared to Location A which is an individual house and where only around 15+ networks were observed within range.

Thus by merely taking the Beacon frames and extracting ssid and signal strength information from them, anyone can identify the network that is present at any house. The stronger the signal, the nearer is the network, to where the data is being captured.

From this point on, only data from Location A is used in the thesis for the sake of brevity. To verify the integrity of the experiment, data from Location B was captured for 3 weeks and the information presented in table 5.3 on page 22 is extracted from the collected data. Thus verified, we would concentrate on the primary target of our analysis and testing, Location A.

5.3 Identifying the Devices belonging to a Network

From the 802.11 header that we observe from this point, we would be able to see the MAC addresses of the devices that are involved in the transmission of each packet. Every device in the network is identified within its LAN using the MAC address. This address is the address of the Network Interface Controller (NIC) that is present in each device; devices communicate over the network via the network interfacing hardware present in them. This MAC address is provided by the manufacturer of the NIC and is unique globally; it is very rarely changed.

From the MAC addresses that we collected at this point, we can get a list unique MAC addresses, each corresponding to a single device (two different MAC addresses may represent a single device if two interfaces from the same device are online on the same medium). From the data collected, we observed around 250+ unique MAC addresses from Location A. Assuming some of the devices might have more than one interface over the wireless medium, we can approximate the value to 200. Thus for approximately 15 networks, there are 200 devices; hence around 13 devices per home. This clearly illustrates the ongoing IoT phase. Earlier, home networks used to have

Table 5.3: Location B - List of Networks

Network	Signal Strength (dBm)
975D6C	-90.05
B6C5E0	-90.13
mae1934	-80.57
Niecy86	-87.32
4E1854	-91.08
2CB03A	-89.02
2E805A	-68.33
0236E0	-89.17
You want Internet or nah?	-90.60
FE2678	-88.64
44931E	-88.33
maverick	-89.73
optimumwifi	-88.69
optimumwifi_Passpoint	-88.40
DIRECT-Ng	-85.12
edshah_guest	-55.81
AA68B8	-89.58
174D8C	-79.11
edshah	-56.47
9C86F4	-90.75
NeruppuDa	-35.04
A047B4	-88.89
D98364	-90.00
GalaxyOne	-90.90
110898	-81.00
WiFi Hotspot 2323	-87.43
bell44	-88.71
Sameer	-83.96
Batterbee	-88.63
NETGEAR86	-89.30
7ca4aa	-89.88
LA ISLA PR	-90.21
f4c2aa	-88.32
B57710	-80.48
T-Mobile Broadband52	-88.11

1 or 2 laptops and 1 or 2 mobile phones connected to the Internet. Now every house hold easily has 10+ devices connected online and it only keeps increasing.

In the previous section, we saw that Beacon frames contain SSID information and are transmitted by the APs. Hence the source address in a Beacon Frame would be the MAC address of an AP. From this information, we can get the list of APs that belong to each network. To identify which AP a device belongs to, we have to

examine the 4 address fields present in 802.11 packets namely Transmitter, Receiver, Source and Destination along with the DS flag present under 'Frame Control'. Based on the direction of the packet transmission, one of the 4 address fields would have the MAC address of the AP.

Table 5.4: IEEE 802.11 - Values in Address Fields based on the DS Value

ToDS	FromDS	Address 1	Address 2	Address 3	Address 4	Interpretation
0	0	RA=DA	TA=SA	BSSID	NA	Packet transmitted within the BSS; Both Transmitter and Receiver belong to the AP
1	0	RA=DA	TA=BSSID	SA	NA	Packet comes from a device outside BSS; Receiver belongs to AP
0	1	RA=BSSID	TA=SA	DA	NA	Packet is meant for a device outside BSS; Transmitter belongs to AP
1	1	RA=BSSID	TA=BSSID	DA	SA	Packet relayed between two APs;

From the information available from table 5.4 on page 23, we can associate a device with the AP it belongs to. Using the list of network names and their APs got previously and this list of APs and their devices, we can find out which network each device belongs to.

Thus anyone passively observing the traffic from outside can get a list of APs present under each network and then the devices present under each AP and associate the device and the network name to get the list of devices present under each network.

5.4 Identifying the Type of Device and Vendor using OUI

Every MAC address is composed of 2 parts - OUI and the serial number. OUI is the Organizationally Unique Identifier and identifies a manufacturer or vendor or other organizations. According to IEEE, vendors or manufacturers are called assignees and they buy the OUI from the IEEE Registration Authority. OUI is 24 bits long and constitutes half of a MAC address, the first 3 octects. The serial number is also 24

bits long and identifies a device uniquely. Hence a MAC address = OUI + serial number.

Since we have the MAC addresses from the packet headers that we have collected, we would be able to translate the OUI to identify the vendor. Tools like Wireshark readily do it and give the resolved MAC address. Just by looking at the vendor or the manufacturer, we would be able to say what kind of device it is most of the time.

Table 5.5 on page 24 lists a set of OUIs collected from the data that we captured and the assignees that bought them or it belongs to.

Table 5.5: Example List of OUIs and Assignees

OUI	Assignee
58-E2-8F	Apple, Inc.
9c-f3-87	Apple, Inc.
40-98-ad	Apple, Inc.
00-0f-00	Legra Systems, Inc.
c0-bd-d1	SAMSUNG ELECTRO-MECHANICS(THAILAND)
98-ca-33	Apple, Inc.
18-b4-30	Nest Labs Inc.
64-b5-c6	Nintendo Co.,Ltd
f8-04-2e	SAMSUNG ELECTRO-MECHANICS(THAILAND)
b0-ee-7b	Roku, Inc.
14-b7-f8	Technicolor CH USA Inc.
c0-56-27	Belkin International Inc.
28-56-5a	Hon Hai Precision Ind. Co.,Ltd.

Hence just by using any widely available tool like Wireshark, anyone passively observing the traffic from outside would be able to say what kind of devices are present in a network.

Since we already know which devices belong to each network from the previous section, we can resolve the MAC addresses of those devices and we would know exactly the type of devices present in each network.

Table 5.6 on page 25 shows a subset of the list of devices belonging to network 'HostNetwork' at Location A after OUI resolution. The Device Type is the actual type of the device; this is not taken from the analysis but given for reference. Just by

Table 5.6: Location A - Subset of List of devices after OUI resolution

Network Name	Access Point Address	Device Address	Device Type*
HostNetwork	Apple_ea:b3:5e	NestLabs_12:05:00	Nest Thermostat
		NestLabs_13:7a:55	Nest Thermostat
		NestLabs_14:48:4b	Nest Thermostat
		NestLabs_14:58:68	Nest Thermostat
		NestLabs_15:3c:6e	Nest Thermostat
		NestLabs_2b:2b:61	Nest Thermostat
		Nintendo_09:28:c7	Nintendo Console
		Nintendo_fa:9d:30	Nintendo Console
		SamsungE_e1:c2:0a	Samsung TV
		Legra_43:46:83	Raspberry Pi Device
		HonHaiPr_08:bf:14	Printer
		Apple_50:6a:f1	Apple TV
		Apple_1c:ce:28	Laptop
		Roku_a4:6d:16	Roku
		WistronN_34:2d:38	Washing Machine
		WistronN_34:2d:61	Dryer

** Device Type is not inferred from the Analysis*

looking at the resolved Device Address we can learn or at least guess the type of the device. For example, Nest Thermostats - their OUI is resolved to NestLabs and from that information we can say it must be either a thermostat or a doorbell or camera or one of the IoT devices provided by NestLabs and hence an IoT device. From the OUI resolution of Samsung TV, Apple TV and Laptop, we can guess it must a device from the manufacturer concerned i.e, they must be either a smart TV or a smart phone or Ipad or Macbook etc., and hence most probably a non-IoT device. The same is the case for Nintendo console and Roku too. The washing machine and the dryer are not that easy to guess. But we can definitely say that there are 6 Nest devices, 1 Samsung device, 1 Roku, 2 Nintendo consoles, 10+ Apple devices along with some other devices in network 'HostNetwork' at Location A just by looking at the resolved MAC addresses.

Thus combining all the information got from previous sections, we can conclude that anyone passively observing the traffic from outside the network can guess the network name of the house he is collecting data from, find out the list of devices

present in the network and also know the number of each type of device present in that house.

5.5 Degree of Activeness of a Device

IoT devices are often small devices and are battery powered. Operating on battery serves a number of purposes including ease of installation anywhere, continued operation even when the power supply is cut off (important for home applications like Smart Doors and Smart Window systems) and also renders an aesthetic value. Since they operate on battery, they try to conserve as much power as possible. Their hardware is also designed in such a way so as to consume as less power as possible; they typically have low power processors, small memory, lower storage capacity etc., - resources that are just enough to perform the one task they are designed to do. On the other hand, devices like laptops and smart phones do not have these constraints. Even though they do have battery, its capacity is much bigger than that of IoT devices and can be charged regularly (typically designed to last for a whole day before recharging again). They also perform a variety of tasks like surfing the Internet, playing media files, making calls, sending and receiving messages etc., Hence their system resources are also quite high to cope up with performing a variety of intensive tasks.

Devices go into Power Save mode after sending a signal to the router that they are going to Sleep. This is done by sending a packet called NULL frame, which is a subtype of Data Frames. NULL frames are exclusively sent from the clients to the Access Point. When the client is ready to go to Sleep, it sends a NULL frame to the Access Point with the PWR MGT flag set to 1. On receiving it, the AP will understand that the client is going to Sleep and will start buffering the packets that are meant for the client until the client wakes up. Clients wake up, either when they receive a stimulus from the environment (eg: Smart Light bulbs going to Sleep when there is no motion detected in the environment and coming back to operation when

there is movement in the vicinity) or by following their internal time clock. Either way, when they wake up, they again send a NULL frame to the Access Point with the PWR MGT flag set to 0. The AP then understands that the device has woken up and resumes sending packets to it. Clients also use NULL frames when they are roaming and are going to switch from one network to the next.

We are going to look at the power management data collected from each device and try to learn about how active a device is over the wireless interface.

Table 5.7: Location A - Network 'HostNetwork' - Comparison of One Day's Duration of Sleep and Active States of an Apple TV, Nest Thermostat and Access Point

Apple_50:6a:f1 (Apple TV)				**NestLabs_15:3c:6e (Nest Thermostat)**				**Apple_ea:b3:5e (AP)**	
Active		**Sleep**		**Active**		**Sleep**		**Active**	
Dur. (secs)	**% of Trans.**	**Dur. (secs)**	**% of Trans.**	**Dur. (secs)**	**% of Trans.**	**Dur. (secs)**	**% of Trans.**	**Dur. (secs)**	**% of Trans.**
3	72	3	69	3	81	3	22	86118	100
6	4	6	4	6	2	6	6		
9	2	9	1	9	1	9	3		
12	5	12	6	12	12	12	4		
24	1	15	1	15	0	15	3		
45	1	27	1	18	1	18	3		
81	5	39	1	21	2	21	4		
102	1	81	5	51	0	51	4		
363	1	93	1	63	0	63	9		
Active > Sleep				Sleep > Active				Only Active	

Dur. - Duration, Trans. - Transitions
Data included is the data from 1 day which was selected randomly

Table 5.7 on page 27 shows the percentage of transitions into Active or Sleep mode for each duration segment for 3 different devices present in Location A under network 'HostNetwork'.

Figure 5.1 on page 28 shows the variations in the degree of two devices - Samsung TV and Nest Thermostat at Location A.

By looking at the power management state data of all devices, we are able to see differences in the duration they are Active, clearly highlighting the different types of devices. Device Apple_ea:b3:5e acts as Access Point for this network. This is

Table 5.8: Location A - Network 'HostNetworkGuest' - Comparison of One Day's Duration of Sleep and Active States of devices Apple_4a:aa:13, Apple_3e:39:83 and Access Point

Apple_4a:aa:13				**Apple_3e:39:83**				**82:5e:b3:ea:96:e0 (AP)**	
Active		**Sleep**		**Active**		**Sleep**		**Active**	
Dur. (secs)	**% of Trans.**	**Dur. (secs)**	**% of Trans.**	**Dur. (secs)**	**% of Trans.**	**Dur. (secs)**	**% of Trans.**	**Dur. (secs)**	**% of Trans.**
3	50	3	65	3	89	3	83	85590	100
6	4	9	8	6	3	6	5		
9	8	21	4	9	2	9	2		
12	4	36	4	12	1	12	2		
15	8	45	4	15	1	15	1		
18	8	315	4	18	< 1	18	1		
36	4	612	4	42	< 1	30	1		
54	4	1338	4	102	< 1	102	< 1		
63	4	4704	4	1014	< 1	1116	< 1		
Sleep > Active				Active ≈ Sleep				Only Active	

Dur. - Duration, Trans. - Transitions
Data included is the data from 1 day which was selected randomly

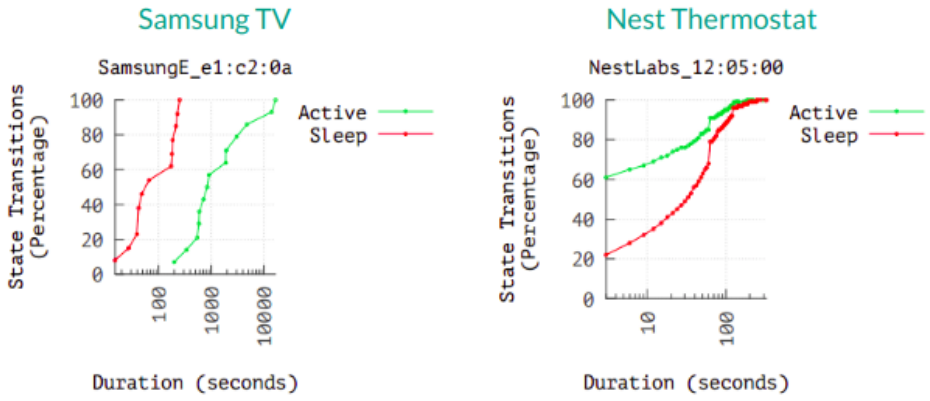

Figure 5.1: Location A - Comparison of Degree of Activeness of Samsung TV and Nest Thermostat - 1 day's data

identified by the fact that it does not send any NULL frames (though one some days it does send - in those periods, it is not acting as the AP); only clients send the NULL frames to Access Points. (It was also identified as the AP simply by looking at the value present in field wlan.bssid in the collected packets). Nest Thermostat devices spend more time in the Sleep mode than in the Active mode. Nearly 50% to 80% or above transitions to Active state lasted for only 3 or less than 3 seconds. On the other hand, only 12% to 25% transitions to Sleep state lasted for 3 or less than 3 seconds. The second most prominent duration in Active state is 9 to 12 seconds which is the duration of nearly 12% to 23% transitions. In sleep mode, the next prominent duration is 60 to 63 seconds where 5% to 14% of the transitions last. This behaviour is typical of IoT devices; they go into Sleep mode as much as possible to conserve power as they are battery operated. Apple TV spent time equally in both the modes. Nearly 69% to 94% transitions spent 3 or less than 3 seconds in both Active and Sleep state. This means it toggles frequently between these 2 states and has not been Active much on the wireless interface. Samsung TV is Active 100% most of the time it is powered ON. Most of the days, no data was collected about the device in Sleep state. This could mean that the device was powered off when not in use most of the time. Hence in comparison, Samsung TV was used more at the site of the data collection than the Apple TV. Also from the collected data, duration of Active state ranges from 2 hours (which is a Tuesday) to 10+ hours (weekends). Roughly speaking, this could be the duration of hours of watching TV and it makes sense that more TV was watched during the weekends than on the weekdays. The Raspberry Pi was active 100% of the time on all days because it was constantly collecting data 24x7. Not much power management information is available about Ipads, Iphone and Nintendo devices on wireless interface; NULL frames from them were not captured in our data collection. This could be because the devices are out of range from the point of our data capture. Also very minimal or no wireless packets

were captured from Insteon Hub, Mac Minis, Laptop, TP-Link, Washing machine and dryer, Printer; this could be because these devices are connected and function actively over the wired interface and do not use wireless interface in general. Roku also behaves the same way; the packets collected from Roku indicate that it is always in Sleep mode over the wireless interface. But it is very Active in the wired interface. Hence it is performing and streaming all the data over the wired interface and not much is happening on the wireless interface. Apple device Apple_de:65:91 exhibits a behaviour similar to that of Apple TV. Consistently, more than 90% of its transitions to both Active and Sleep states lasted for 3 or less than 3 seconds As said before, this means it toggles very frequently between the two states. Apple devices Apple_c7:eb:05 and Apple_61:b1:70 spend slightly more time in the Sleep state than in the Active state, most of the time. This means that they are trying to conserve power, though not as stringently as Nest Thermostat devices and hence they could be considered to fall in the IoT device category. Device Apple_86:34:26 is always in the sleep state. From the data collected, it was never Active. But we find that it is highly Active in the wired interface. Hence, we conclude that it does not actively operate over the wireless interface. Only very minimal data was collected from device Apple_15:b9:21; from the data collected, we see that it spends more time in the Active state than in the sleep state. We can say that the device is switched OFF when not in use and while it is powered ON, it spends more time in the Active state. Hence it falls in the non-IoT device category. Device Apple_0f:fd:22, on most days, spends more time in the Active state, just like Apple_15:b9:21; but on some days, it spends way more time in the Sleep state. This leads to the conclusion that it is not an IoT device to exhibit a consistent power-save-mode behaviour; rather it stays in Active or Sleep mode according to usage. Device Apple_0c:59:e5, which also had data only from a few days, spent time equally in both Active and Sleep State. eGauge devices were not active on the wireless interface; they operated actively over the wired interface.

Let us move on to devices on the network 'HostNetworkGuest'. Access Point for this network is device 82:5e:b3:ea:96:e0. It was identified based on the same grounds as AP of network 'HostNetwork'. Device Apple_4a:aa:13 was more Active on some days and spent more time in Sleep mode on other days. This mixed behaviour is typical of non-IoT devices whose rate of activeness depends purely on their usage. Device Apple_e8:26:91 spends more time in the Sleep state most of the days. But active states do last considerably longer; they are not as short as the ones typical of IoT devices. Apple_3e:39:83 spends time equally in both the states; also, most of the time some 75% to more than 90% of the transitions last for only 3 or less than 3 seconds. That means it toggles quite frequently between the two states. Apart from this, it does not do anything on the wireless interface. For devices Apple_5b:81:a1 and Apple_a5:db:34, data from only a few days were collected, even on the wired interface. This leads to the conclusion that they were switched off when not in use. From the collected data, on some days, they spend more time in the Sleep state and on some days, they spend more time in the Active state. Therefore, they toggle between the two states based on usage and hence, fall under the non-IoT device category. Raspberry Pi device was active 100% of the time. This is expected as the device is continuously collecting data 24x7. One of the Nintendo devices present in network 'HostNetwork', joins the guest network at times. But here also it exhibits the same behaviour. Not much power management information is available on the wireless interface but it is very active on the wired interface. From device SamsungE_16:fb:15, data was collected from only one day. This could be a guest device, a device that joined the network only for that day and never came into this network again. The data shows that 92% of its transitions to both Active and Sleep state last for only 3 or less than 3 seconds, meaning it did not spend longer time in any state; it just toggles between the two states.

Device Apple_39:19:61, belonging to network 'TC8717TA5', spent more time in the Active state on some days and more time in Sleep state on other days. This leads to the conclusion that it toggled between the states according to usage and hence, it is a non-IoT device. Device Apple_e1:32:01, also present on the same network, was always in the Sleep state on the days the data was captured. This means that it was active over the wired interface whenever it was switched ON, as we can see its traffic on wired interface. Devices Advantec_f5:cf:a4, Advantec_f5:d3:a0, Advantec_f5:d3:ce, Advantec_f7:87:c5, Z-Com_a2:a9:34, BelkinIn_a4:a2:fe, HonHaiPr_da:c3:73 and HonHaiPr_fc:3d:62 from networks 165, 167, 166, 170, 161, flynn_Ext, Hightower Power-1 and WIFIFC3D5E respectively had no power management data from them. The absence of NULL frames originating from them as well as the information from their traffic on the wireless interface (from field wlan.bssid) show that they are Access Points.

Hence, just by looking at the power management information from outside the network, we are able to see how active a device is over the wireless interface. Hence, if an intruder wants to identify an Access Point (which is usually the router in home environments), he could just search for devices from which NULL frames do not originate. Once the router is identified, he could use this information to hack it by some means and gain access into the network, monitor all the traffic over the wireless interface as every device in the wireless interface communicates via the router. And just by filtering devices that spend more time in the Sleep state than in the Active state consistently, he can identify IoT devices, devices that do not have robust security and hence, vulnerable and easy to hack.

5.6 How often do Devices change states ?

We saw earlier that IoT devices, since they are battery operated and have minimal resources, try to conserve as much power as possible by going into sleep mode as much

as possible. Non-IoT devices do not have this need as they have greater resources at hand and a large battery and power to use.

Hence, first we will look at how frequently devices toggle between Active and Sleep state.

Also, we will check how consistent are these devices in toggling between Active and Sleep states. If they are IoT devices, they would always be performing a few closely related tasks. Eg: The only job of a smart light is to switch ON whenever it detects any motion in the vicinity and to switch OFF when there is no motion for some period; it may send this data over to a remote server for further processing. Also IoT devices would consistently try to save power as much as possible. Hence the frequency with which they toggle between the states should remain consistent over the days. But non-IoT devices, on the other hand, perform a variety of tasks; they also do not try to go into power save mode as often as the IoT devices. They stay Active as long as they are in usage and that usage varies vastly user to user and from time to time.

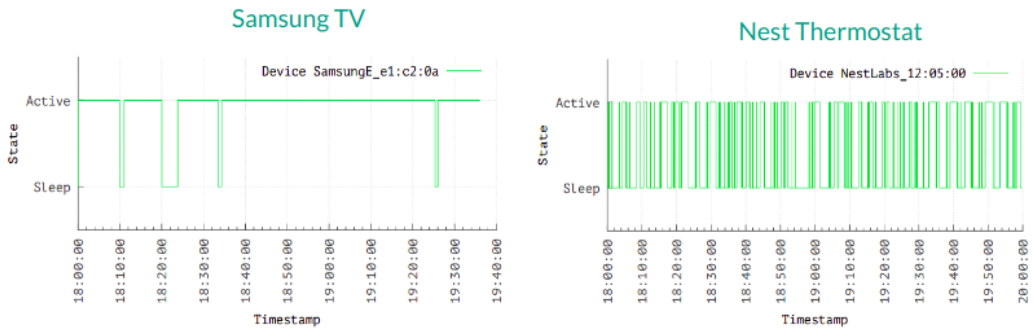

Figure 5.2: Location A - Comparison of Frequency of State Change of Samsung TV and Nest - 2 hour's data

First let us take network 'HostNetwork'. Figure 5.2 on page 33 shows the differences in the rate at which state transitions are made between Samsung TV and Nest Thermostat. Access Point (Router) and Raspberry Pi device make 0 transitions per hour; that means they do not change states at all. The variance for these devices'

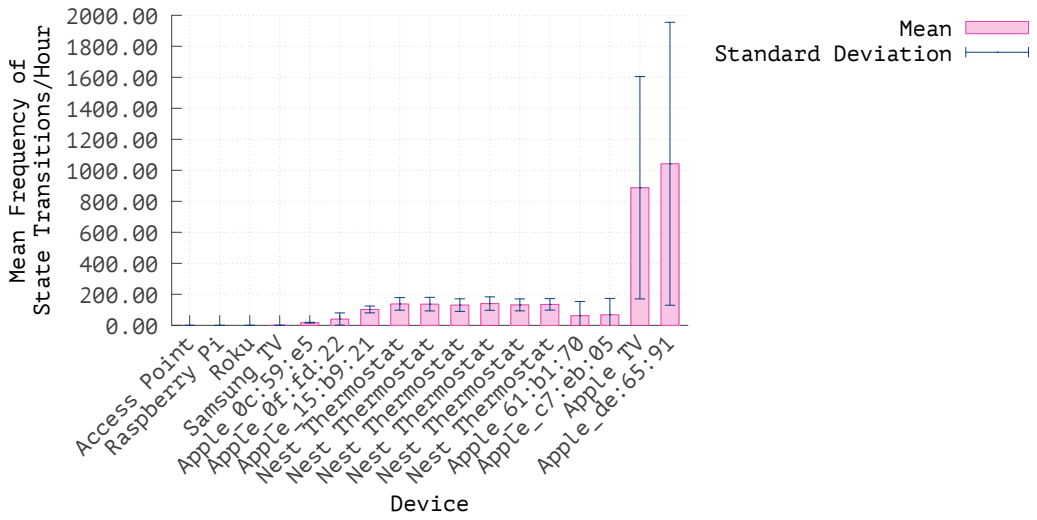

Figure 5.3: Location A - Network 'HostNetwork' - Mean of Number of Transitions per hour and their fluctuation over 120+ days

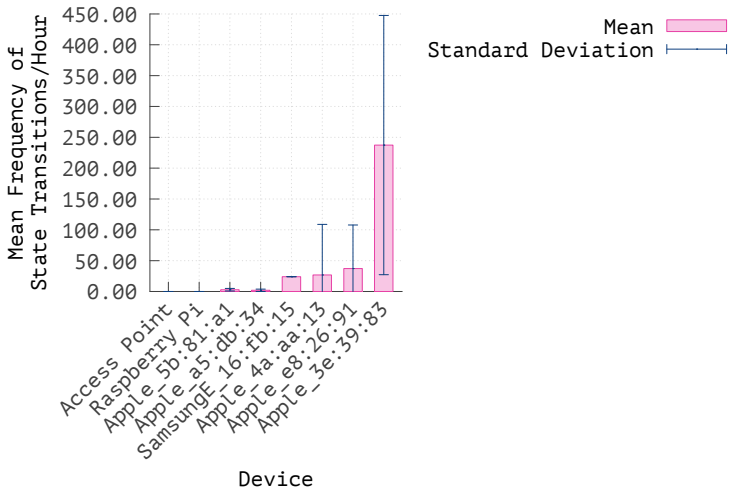

Figure 5.4: Location A - Network 'HostNetworkGuest' - Mean of Number of Transitions per hour and their fluctuation over 120+ days

Table 5.9: Location A - Network 'HostNetwork' - Inference - Mean of Number of Transitions per hour and their fluctuation 120+ days

Device	Mean	Variance	Comments
Access Point (ea:b3:5e)	0.0	0.0	Zero Transitions, Zero Variance, Always Active
Raspberry Pi 1 (35:bb:d5)	0.0	0.0	Zero Transitions, Zero Variance, Always Active
Raspberry Pi 2 (43:46:83)	0.0	0.0	Zero Transitions, Zero Variance, Always Active
Roku AP (72:1a:5f)	0.0	0.0	Zero Transitions, Zero Variance, Always Sleep
Roku_a4:6d:16	0.0	0.0	Zero Transitions, Zero Variance, Always Sleep
Samsung TV (e1:c2:0a)	0.03	0.07	Negligible mean, Negligible Variance, Active most of the time over all days
Printer (08:bf:14)	2.0	2.0	Very small mean and variance, only few days' data available
Apple_0c:59:e5	17.23	6.3	Very small mean, Very small variance, Only few days' data available
Apple_0f:fd:22	40.44	1543.23	Small mean, Comparatively small variance, More time in a state according to usage
Apple_15:b9:21	102.62	488.28	Comparatively small mean and variance, More time in Active state
Nest Thermostat (2b:2b:61)	138.35	1658.62	Comparatively small mean and variance More time consistently in Sleep state over the days
Nest Thermostat (12:05:00)	136.94	1935.57	Comparatively small mean and variance More time consistently in Sleep state over the days
Nest Thermostat (13:7a:55)	130.83	1644.26	Comparatively small mean and variance More time consistently in Sleep state over the days
Nest Thermostat (14:58:68)	140.46	1858.48	Comparatively small mean and variance More time consistently in Sleep state over the days
Nest Thermostat	132.11	1493.21	Comparatively small mean and variance More time consistently in Sleep state over the days
Nest Thermostat	135.38	1382.57	Comparatively small mean and variance More time consistently in Sleep state over the days
Apple_61:b1:70	61.95	8443.91	Small mean and comparatively high variance, More time in Sleep state but frequency of transitions fluctuates a lot Not as stringent as Nest Thermostat in conserving power
Apple_c7:eb:05	68.3	11125.83	Small mean and comparatively high variance, More time in Sleep state but frequency of transitions fluctuates a lot Not as stringent as Nest Thermostat in conserving power
Apple TV (50:6a:f1)	887.98	514256.0	Very high mean frequency and variance Just toggles between the two states Not much activity in wireless interface
Apple_de:65:91	1042.39	833176.83	Very high mean frequency and variance Just toggles between the two states Not much activity in wireless interface

Table 5.10: Location A - Network 'HostNetworkGuest' - Inference - Mean of Number of Transitions per hour and their fluctuation over 120+ days

Device	Mean	Variance	Comments
82:5e:b3:ea:96:e0	0.0	0.0	Zero Transitions, Zero variance, Always active
Raspberry Pi 1 (35:bb:d5)	0.0	0.0	Zero Transitions, Zero variance, Always active
Apple_a5:db:34	1.95	3.68	Small mean and variance, Longer duration in a state and consistent over the days
Apple_5b:81:a1	2.9	4.47	Small mean and variance, Longer duration in a state and consistent over the days
SamsungE_16:fb:15	24.0	0.0	Small mean and no variance, Data from only one day, Probably a guest device
Apple_4a:aa:13	26.79	6729.07	Small mean, High variance, Longer duration in a state, but duration fluctuates greatly
Apple_e8:26:91	37.19	4979.43	Small mean, High variance, Longer duration in a state, but duration fluctuates greatly
Apple_3e:39:83	237.4	44192.66	High Mean, Very High Variance, Frequently toggles between the two states, Fluctuates greatly in duration

mean frequency over a period of 120+ days is zero, meaning their behaviour is exactly the same over all those days. This is reasonable as we saw earlier that they always stay in the Active mode (to serve other devices and to collect traffic data respectively) and never transition to sleep mode. Roku also made zero transitions and its variance over all days is also 0. Since it is not active over the wireless interface, it always stays in sleep mode. The mean frequency of transitions made by Samsung TV is very small. This means most of the time, it stayed in one state and transitioned only rarely. It's variance is also very negligible. This is in accordance with our observation before where Samsung TV stayed in Active mode most of the time most of the days. Devices Apple_0c:59:e5 and Apple_0f:fd:22 have lower frequencies of transition, meaning they spend longer duration in a state before transitioning to the next. Nest Thermostats have a higher frequency of transitioning than the devices mentioned so far. This is in par with our previous observation; Nest Thermostats go into Sleep mode as much as possible to conserve energy; they do not spend longer duration in Active state. And their variance is also comparatively low; this means they exhibit this behaviour consistently over these 90+ days. Devices Apple_61:b1:70 and Apple_c7:eb:05, though their mean frequency is less than that of Nest Thermostat devices, make transitions

that fluctuate greatly over the days; variance is higher. As said before, though they try to conserve power, they do not do it as stringently and consistently as the Nest Thermostat devices. The mean frequency of Apple TV and Apple_de:65:91 show that they keep toggling between the Sleep and Active states very frequently. This is in accordance with our observation from the previous section - their duration in both Active and Sleep states lasted for 3 or less than 3 seconds most of the time.

Second, let us take network 'HostNetworkGuest'. Access Point and Raspberry Pi devices make 0 transitions per hour. Because, as we saw before they are always up and running. Their variance is also zero. Hence, this behaviour is consistent over the 90+ days. Less data was collected from Apple_5b:81:a1 and Apple_a5:db:34 compared to the other devices. From the data collected, their mean frequency and variance are low. This means they spend a lot of time in a state before transitioning to the next and this is consistent over the days (for the days we collected the data). Devices Apple_4a:aa:13 and Apple_e8:26:91 have lower frequencies of transition. Hence they spend considerably longer duration in any state. Their variances are high; hence the duration they spend in a state fluctuates considerably over the days. As discussed before, they must fall under the non-IoT device category as IoT devices would be consistently spending more time in the Sleep state. Device Apple_3e:39:83 has a very high mean frequency of transition and variance. This adds strength to our previous observation that it keeps toggling between the two states with most of its transitions lasting for 3 or less than 3 seconds.

Since we are observing the data in monitor mode, we were able to capture data from devices that were not present in the networks 'HostNetwork' and 'HostNetworkGuest'. Simply all packets within the observable range were captured. Device Apple_39:19:61 belonging to network 'TC8717TA5' has a mean frequency of 7.0 transitions per hour; its variance is mid-range about 108.1. This means that they spend longer duration in a state and the duration varies over the days. Varying duration as

well as longer duration in Active state over the Sleep state leads to the conclusion that this device is a non-IoT device. Device Apple_e1:32:01, also belonging to the same network, has zero mean and zero variance; this means that it always stayed in one state and never transitioned to the other state. From the data from previous section, we know that it always stayed in Sleep state. Hence this result is consistent with that observation. Devices Advantec_f5:cf:a4, Advantec_f5:d3:a0, Advantec_f5:d3:ce, Advantec_f7:87:c5, Z-Com_a2:a9:34, BelkinIn_a4:a2:fe, HonHaiPr_da:c3:73 and HonHaiPr_fc:3d:62 from networks 165, 167, 166, 170, 161, flynn_Ext, Hightower Power-1 and WIFIFC3D5E respectively all had zero mean and zero variance; no data was received from them. This yet again proves that these devices are Access Points of their respective networks.

Hence by combining the information from both how frequently the devices change states and how active the devices are, an intruder can easily identify access points and IoT devices, without even joining the network. For access points, since NULL frames do not originate from them, their mean frequency of transitions per hour (or in a day) and variance will be zero. For devices that are in Active or Sleep mode all the time would also have the same result with regards to mean frequency and variance, but they will have NULL frames originating from them. This would easily distinguish an Access Point from a device that is 100% either active or asleep. Devices that have a comparatively mid-range mean and variance are devices that spend a considerably long duration in a state and are consistent in that behaviour over the days. Data from the previous section will give information as to whether that state is Active or Sleep; devices spending more time asleep are the IoT devices. The rest of the devices, those that either spend more time in the Active state or is not consistent in the duration they spend in the states, would be non-IoT devices.

5.7 Traffic associated with a device

Network Traffic can be defined as the amount of data that is circulating in a network. The various devices present in the network contribute to this network traffic by either sending or receiving data via the network. The basic unit of network traffic is packets. While the header length across various layers of the network remain the same (sometimes it may vary with the inclusion of optional fields), the payload or the actual data that is transmitted or received varies from device to device, task to task. And this traffic is measured in bytes. We get this information from the length of the packets traveling across the network. This information is available in the packet header.

By device traffic, we mean the data that is either sent or received by a particular device; hence the total traffic sent and received by every device constitute the network traffic. While IoT devices are designed for a specific purpose and they usually do one specific task or tasks related to that specific purpose, non-IoT devices are designed to perform a variety of tasks. Therefore, the volume of device traffic for an IoT device must be significantly lesser than that of a non-IoT device; because IoT-devices would be generating traffic from or for a particular task whereas non-devices will be doing the same for a variety of tasks. This difference between the nature of the tasks among the two types of devices not only varies the volume of traffic but also the type of traffic.

5.7.1 Content Type of Packets associated with a Device

Since we are observing the packets from outside without joining the network, we will not be able to get any information regarding the content of the packet or the type of protocol that is used. Without the protocol information, it is difficult to pinpoint exactly what these devices are doing. But externally, we have another information available that can be used for this purpose to an extent - type of packet sent.

Wireless traffic can be categorized into 3 types of packets according to the IEEE 802.11 specification. They are,

- **Data:** Data frames are used to carry the actual data from the upper layers.
- **Control:** Control frames are used to deliver the data by handling inherent issues in wireless networks like collision avoidance and Hidden Node problems.
- **Management:** Management frames are used to maintain the wireless network. Devices join and leave the network, get authenticated etc., all using Management frames.

We are going to focus solely on the data packets now. Data packets have subtypes within them namely - Data(Type 32), NULL(Type 36), Data+Contention Free Acknowledgement(Type 33), Data+Contention Free Poll(Type 34) etc., All these subtypes serve different purposes in transmitting the data. For example, subtype Data packets(Type 32) carry the actual data, NULL frames(Type 36) are used to send power management information to the Access Point based on which the AP either sends the data or buffers it until the device wakes up again. Hence, by looking at the type of packets sent by each device, we will be able to get an idea of the type of content that is actually being carried in the packets merely by observing them from outside the network.

Table 5.11 on page 41 shows the composition of subtypes of packets under the Type Data packets that are transmitted from and received by each device under network 'HostNetwork'.

Of all the subtypes under Type Data, we only see subtypes 32 and 36 involved. Access Point transmits 99.75% of subtype 32 packets, i.e., purely data and only very negligible subtype 36, i.e., NULL frames. This goes to show that most of the time it acts as an Access Point without any NULL frames originating from it. Very rarely it sends NULL frames; during those times it was not acting as an Access Point. Nintendo devices transmit subtype 32 packets all the time; they do not send NULL

Table 5.11: Location A - Network 'HostNetwork' - Average Percentage of Types of Packets transmitted and received in a day

Device	Transmitted		Received	
	Percentage of SubType 32 (%)	Percentage of SubType 36 (%)	Percentage of SubType 32 (%)	Percentage of SubType 36 (%)
Access Point	99.75	0.24	7.77	92.22
Nest Thermostat (2b:2b:61)	0.0	100.00	0.0	0.0
Nest Thermostat (12:05:00)	0.0	100.00	0.0	0.0
Nest Thermostat (13:7a:55)	0.0	100.00	0.0	0.0
Nest Thermostat (14:58:68)	0.0	100.00	0.0	0.0
Nest Thermostat (14:48:4b)	0.0	100.00	0.0	0.0
Nest Thermostat (15:3c:6e)	0.0	100.00	0.0	0.0
Apple TV	0.0	100.00	0.0	0.0
Samsung TV	0.0	100.00	0.0	100.00
Roku AP (72:1a:5f)	100.0	0.0	0.0	100.00
Roku Device (a4:6d:16)	0.0	100.00	0.0	0.0
Laptop (1c:ce:28)	0.0	100.00	0.0	100.00
Printer (08:bf:14)	0.0	100.00	0.0	0.0
Nintendo (09:28:c7)	100.00	0.0	100.00	0.0
Nintendo (fa:9d:30)	100.00	0.0	100.00	0.0
Raspberry Pi	0.0	100.00	0.0	100.00
Apple_de:65:91	0.0	100.00	0.0	100.00
Apple_c7:eb:05	0.0	100.00	0.0	100.00
Apple_86:34:26	0.0	100.00	0.0	100.00
Apple_61:b1:70	0.0	100.00	0.0	0.0
Apple_15:b9:21	0.0	100.00	0.0	100.00
Apple_0f:fd:22	0.0	100.00	0.0	100.00
Apple_0c:59:e5	0.0	100.00	0.0	0.0

This average is the average over 120+ days

Table 5.12: Location A - Network 'HostNetworkGuest' - Average Percentage of Types of Packets transmitted and received in a day

Device	Transmitted		Received	
	Percentage of SubType 32 (%)	Percentage of SubType 36 (%)	Percentage of SubType 32 (%)	Percentage of SubType 36 (%)
Access Point	97.7	2.29	1.2	98.79
Apple_4a:aa:13	0.0	100.00	0.0	0.0
Apple_e8:26:91	0.0	100.00	0.0	0.0
Apple_3e:39:83	0.0	100.00	0.0	100.00
Apple_5b:81:a1	0.0	100.00	0.0	100.00
Apple_a5:db:34	0.0	100.00	0.0	0.0
Raspberry Pi	0.0	100.00	0.0	100.00
Nintendo (fa:9d:30)	100.00	0.0	100.00	0.0
SamsungE_16:fb:15	0.0	100.00	0.0	0.0

This average is the average over 120+ days

frames at all. Rest of the devices like Nest Thermostats, Roku, Laptop, Printer, Raspberry Pi and Apple devices send NULL frames 100% of the time. This means, other than sending power management information (i.e., if they are awake or asleep) to the AP, they do not do much on the wireless interface. No information is available about MacMinis, Washing Machine and Dryer, Insteon Hub etc., No packets were captured from them over the wireless interface at this point. This shows that they are either active only over the wired interface or we are too far from the devices to capture their wireless packets from the air. Only very minimal packets were captured wirelessly from Ipads and a Laptop. But it is impossible that they did not generate any wireless traffic. Hence we conclude that the packets sent by those devices were out of range to capture from our point of data capture.

Coming to the received packets, the access point received both Data packets and NULL frames. 7.77% of the time, it received Type 32 packets and received Type 36 packets 92.22% of the time. Therefore it receives type 36 packets sent by the rest of the devices most of the time. Nest Thermostat devices did not receive any type of packets. This is reasonable because there is no reason for any device to contact the Nest Thermostat nor for the Nest Thermostat to receive any data from any other device. All it has to do is to collect data about the environment (which it will do using the on-board sensors) and send them to the remote server. It is assumed at this point that the collected data is sent via the wired interface. Hence it does not receive any type of packets. Apple TV does not receive any type of packet and Samsung TV receives only type 36 packets. This goes to show that all their streaming and receiving content happens over the wired interface. Nintendo devices, just like their transmission, received only Data subtype of packets. Rest of the devices either receive nothing or receive only type 36 frames.

Table 5.12 on page 41 shows the composition of subtypes of packets under the Type Data packets that are transmitted from each device under network 'HostNet-

workGuest'. Here also, we find only subtypes 32 and 36. The AP of the guest network has 97.7% packets to be of subtype 32, i.e., data packets and 2.29% NULL frames. Hence, like the AP of the main network, this device acts as an AP most of the time by not sending any NULL frames and sends them only rarely. During those times, it could not have functioned as an AP. One of the Nintendo devices from the main network joins the guest network sometimes. Here also, it sends only data packets and no NULL frames. Rest of the devices namely Raspberry Pi, Samsung and Apple devices all send only NULL frames to the AP and do not send any other data. Hence, these devices are not active over the wireless interface other than sending power management information.

Coming over to the received packets, the AP of the guest network also shows the same result as the AP of the main network. It receives only 1.2% of Type 32 packets and 98.79% of type 36 packets. The Nintendo device, as it behaved in the main network, receives only Data subtype packets. Rest of the devices either receive nothing or receive only NULL frames. This leads to the conclusion that these devices, other than letting the AP know whether they are asleep or active, do not do anything else on the wireless interface. It is assumed at this point that Raspberry Pi must be active over the wired interface. It must be sending the collected data over the wired interface. The Samsung and other Apple devices, since we don't know their type, could be active over the wired interface or they could be out of range from the point of data collection or they could simply be inactive.

Table 5.13 on page 44 shows the composition of the types of packets that are transmitted and received by devices from networks other than the ones present at the house where we are capturing the data; they are present in the vicinity, may be in the houses nearby. All the Access Point devices send only subtype 32 frames. For the APs, this behaviour is expected as NULL frames originate from the client devices and are received by the APs. For other devices that are not access points,

we have managed to capture only type 36 packets. On the receiving end, only AP Technico_07:d7:ab has received NULL frames, only NULL frames. For other devices, we do not have any data about the reception of packets. Since these are devices that are from networks that are a little distant from the point of our data capture, may be these devices were out of range from the point of data collection for us to capture packets that were sent to these devices. We were able to capture NULL frames transmitted by them because those were sent to APs and the APs were within our range as we were able to capture packets transmitted from them.

Table 5.13: Location A - Other Networks - Average Percentage of Types of Packets transmitted and received in a day

Network	**Device**	**Transmitted**		**Received**	
		% of SubType 32	**% of SubType 36**	**% of SubType 32**	**% of SubType 36**
TC8717TA5	Apple_02:f4:2d	0.0	100.00	0.0	0.0
TC8717TA5	Apple_13:40:f3	0.0	100.00	0.0	0.0
TC8717TA5	Apple_28:8f:31	0.0	100.00	0.0	0.0
TC8717TA5	Apple_39:19:61	0.0	100.00	0.0	0.0
TC8717TA5	Apple_e1:32:01	0.0	100.00	0.0	0.0
TC8717TA5	AP (Technico_07:d7:ab)	100.00	0.0	0.0	100.00
165	AP (Advantec_f5:cf:a4)	100.00	0.0	0.0	0.0
167	AP (Advantec_f5:d3:a0)	100.00	0.0	0.0	0.0
166	AP (Advantec_f5:d3:ce)	100.00	0.0	0.0	0.0
170	AP (Advantec_f7:87:c5)	100.0	0.0	0.0	0.0
flynn_Ext	AP (BelkinIn_a4:a2:fe)	100.00	0.0	0.0	0.0
Hightower Power-1	AP (HonHaiPr_da:c3:73)	100.00	0.0	0.0	0.0
WIFIFC3D5E	AP (HonHaiPr_fc:3d:62)	100.00	0.0	0.0	0.0
Unknown NW 4	Motorola_fa:89:3e	0.0	100.00	0.0	0.0
Unknown NW 4	Motorola_1d:d1:10	0.0	100.00	0.0	0.0
Unknown NW 6	Apple_ea:04:04	0.0	100.00	0.0	0.0
Unknown NW 6	Apple_a4:11:69	0.0	100.00	0.0	0.0
Unknown NW 6	Apple_0e:38:9c	0.0	100.00	0.0	0.0
Unknown NW 8	6a:31:96:8c:d9:fc	100.00	0.0	0.0	0.0
Unknown NW 8	82:e2:f7:a6:0d:bc	100.00	0.0	0.0	0.0
Unknown NW 16	AP (SenaoNet_11:c4:ee)	100.00	0.0	0.0	0.0

Hence, by just capturing the packets from outside the network without joining it and by reading the type and subtype values in the packet, an intruder will be able to get a rough idea of the contents and purpose of the packet. He will be able to identify

the Access Point (which is also the Router in home networks) by identifying the device that transmits data frames almost always and negligible or no NULL frames.

5.7.2 How much Traffic do Devices Send and Receive ?

Now that we have established the type of content transmitted by the devices, let us take into consideration the volume of traffic that is associated with each device. Volume of traffic can be a key indicator in identifying the type of device. Devices like laptops, TVs, Roku etc., must have a high volume of traffic as they will be used for a variety of purposes like streaming video, playing music, browsing the internet, running software applications that interact with the cloud etc., IoT devices, on the other hand, will not incur a very high traffic as their only purpose is to collect data about the environment and send it to a remote server. Hence, volume of traffic observed externally could be a very good indicator of the type of device unless the device is also connected via the wired medium and is using that over the wireless medium for its activities.

Along with examining the traffic volume that each device is associated with, we would also try to group the devices that are similar in their traffic volumes. Doing this for every parameter that we are analyzing would give us a unique profile about each device when combining all the parameters.

Table 5.14: Traffic Volume Limits for Traffic Categorization

Range (Bytes)	**Traffic Category**
0 - 100	Very Low
101 - 500	Low
501 - 1000	Mid-High
1001 - 5000	High
5001 - 10000	Very High
10001 and above	Extremely High

Table 5.14 on page 45 shows the categories that we are going to divide the devices into, based on their traffic volume. This volume includes traffic that is both sent and received by the device.

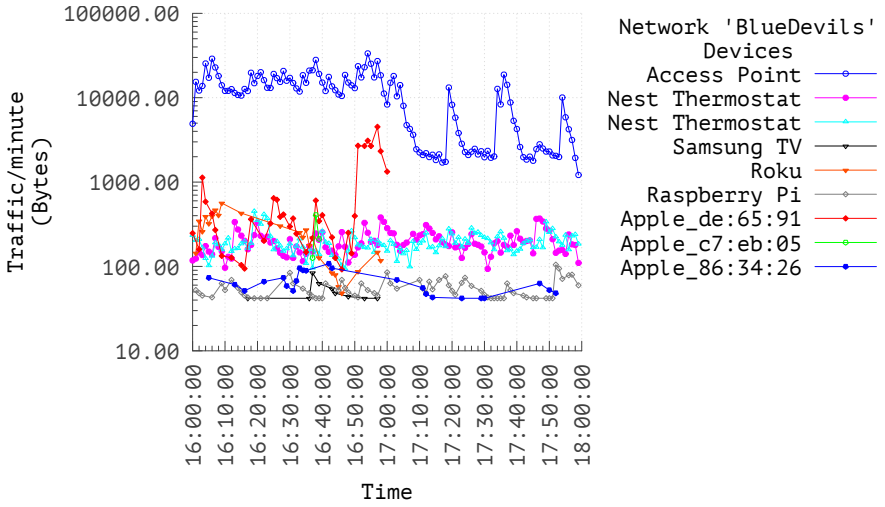

Figure 5.5: Location A - Device Traffic in a minute - Sent and Received - 2 hours

Figure 5.5 on page 46 shows a sample traffic of subset of devices present in network 'HostNetwork' at Location A for a duration of 2 hours. From the figure, we are able to see that the device with the highest traffic is the Access Point. Every device, in order to communicate on the wireless interface, has to communicate with the Access Point (Router), which relays the information to the destination device. Hence, in any given network, it is the Access Point (or the router) that would have the highest traffic on the wireless interface. Device Apple_de:65:91 has highly fluctuating traffic ranging from 100 bytes to more than 5000 bytes (this does not mean that the device exhibits this behaviour at all times) and stops abruptly mid-way at around 5pm. This could be the time the device was switched off. We can see that the Nest Thermostat devices resemble very closely in their traffic volumes as they both carry out the exact same functionality. We see that the device Roku has a lower traffic volume than that of the device Apple_de:65:91 in this time frame and its traffic also abruptly stops at around 5pm. May be Roku was also switched off at the same time. Raspberry Pi's traffic

Table 5.15: Location A - Network 'HostNetwork' - Mean of Average Traffic generated in a minute over all days

Device	Avg. of Mean(Bytes)	Traffic Category
Laptop (1c:ce:28)	55.93	Very Low
Apple_86:34:26	62.4	Very Low
Samsung TV	63.43	Very Low
Raspberry Pi	90.18	Very Low
Apple_0c:59:e5	171.5	Low
Apple_61:b1:70	366.19	Low
Nest Thermostat (12:05:00)	303.89	Low
Nest Thermostat (13:7a:55)	313.47	Low
Nest Thermostat (14:48:4b)	323.65	Low
Nest Thermostat (14:58:68)	336.58	Low
Nest Thermostat (15:3c:6e)	330.45	Low
Nest Thermostat (2b:2b:61)	326.54	Low
Apple_c7:eb:05	569.63	Mid-High
Apple_0f:fd:22	792.37	Mid-High
Apple_15:b9:21	627.2	Mid-High
Nintendo (09:28:c7)	641.01	Mid-High
Roku Access Point	918.39	Mid-High
Roku Device	924.62	Mid-High
Nintendo (fa:9d:30)	1231.17	High
Apple TV	1769.52	High
Apple_de:65:91	4927.26	High
Router	17442.26	Extremely High

To get this Average of Mean value, first traffic per minute was calculated and its mean over a day is calculated. Then an average of all the mean traffic per minute per day day is taken. Average is over 120+ days

is very minimal. This is in accordance with our observation from previous section where we saw it sent only NULL frames and did not do anything else. Most probably, the device is sending the collected data to the remote server via the wired interface. We see that the Samsung TV is operated for an hour from roughly 4.10pm to 5 pm. All these streaming devices - Roku, Samsung TV and Apple_de:65:91 (possibly a streaming device because of its high traffic) switching off at around the same time could mean that the occupants are leaving the house. This is an illustration of the traffic volume of various devices over a duration of 2 hours.

From table 5.15 on page 47, we can clearly see the differences in the traffic volume of the various devices present under nerwork 'HostNetwork'. Devices Apple_86:34:26,

Table 5.16: Location A - Network HostNetworkGuest - Mean of Average Traffic generated in a minute over all days

Device	Avg. of Mean(Bytes)	Traffic Category
Raspberry Pi	80.73	Very Low
Apple_e8:26:91	153.31	Low
Apple_5b:81:a1	126.75	Low
Apple_a5:db:34	119.16	Low
Access Point	854.78	Mid-High
Apple_4a:aa:13	1032.92	High
Apple_3e:39:83	1754.51	High

To get this Average of Mean value, first traffic per minute was calculated and its mean over a day is calculated. Then an average of all the mean traffic per minute per day day is taken. Average is over 120+ days

Laptop, Samsung TV and Raspberry Pi all have very low traffic on the wireless interface. It is unknown what type of device is Apple_86:34:26; so it is difficult to draw a conclusion as to why its traffic is low. The low traffic volume of the Laptop could be because it might have been taken out everyday when the occupants leave home for work and it is not used much when at home. Or it could simply be out of range from the point of data capture. Both the Samsung TV and the Raspberry Pi device are not very active over the wireless network. That's why their wireless traffic volume is low. They must be very much active over the wired interface; let us verify that in the coming sections. The Nest Thermostats all have a similar traffic volume and it is quite low. Considering results from above, it is merely sending power management information over the wireless interface. Devices Apple_0c:59:e5 and Apple_61:b1:70 also have a low traffic volume. From the information above, even these devices transmit only power management frames over the wireless interface. Since they transmit only NULL frames and the traffic volume is very low, it means they send fewer NULL frames; meaning they stay in any state for longer duration. This is in par with our observation from Table 5.9 on page 35 for device Apple_61:b1:70. Nintendo device

has a high traffic; since it is a gaming console it is reasonable that it has a mid-range high traffic. Apple TV and device Apple_de:65:91 also have a high traffic and from previous section we know that these devices sent only subtype 36 packets. This means they sent high amounts of NULL frames. This aligns with our observation that these devices toggle very frequently between the two states without doing anything else on the wireless interface. The Router (Access Point) has the highest traffic among all the devices. This is expected because every device on the wireless interface has to communicate via the router; this is also verified from the previous section where more than 99% of the time, the Access Point sends subtype 32 packets (Data packets), proving that it is relaying the traffic between the various devices under it.

Table 5.16 on page 48 shows the traffic volume data of devices from network 'Host-NetworkGuest'. Raspberry Pi has the lowest traffic among all the devices. This could be because the device is sending all of its collected data over the wired interface; hence its traffic on the wireless interface is very low. All the 3 Apple devices, Apple_e8:26:91, Apple_5b:81:a1 and Apple_a5:db:34 have very low traffic. This leads to the conclusion that they are either not active over the wireless interface and using the wired interface for all their activities or they are devices with low traffic in general. The Access Point has a mid-range traffic. This is justified because the number of devices under it are so few in number and their respective traffic are not very high either. Devices Apple_4a:aa:13 and Apple_3e:39:83 have traffic higher than the Router (AP) because these devices also join networks 'TC8717TA5' and another unknown network respectively and this traffic is the total traffic of the device from all the networks. But a higher traffic indicates that they are most likely non-IoT conventional devices like smartphones or laptops.

From table 5.17 on page 50, we come to information about the traffic volume of devices present under networks that are foreign to the location of the data capture. Most of these devices are Access Points (Routers in a home environment) but still

we find that their traffic volume is quite low; since the location of data capture is an individual home, the location of these Access Points (and in turn the houses they are present) would be quite far from our location and hence only very minimal packets from those networks were captured in monitor mode.

Table 5.17: Location A - Outside Networks - Mean of Average Traffic generated in a minute over all days

Network	Device	Avg. of Mean(Bytes)	Traffic Category
TC8717TA5	Apple_e1:32:01	63.0	Very Low
161	AP (Z-Com_a2:a9:34)	97.55	Very Low
TC8717TA5	AP (Technico_07:d7:ab)	107.87	Low
166	AP (Advantec_f5:d3:ce)	284.0	Low
flynn_Ext	AP (BelkinIn_a4:a2:fe)	162.83	Low
Hightower Power-1	AP (HonHaiPr_da:c3:73)	286.95	Low
TC8717TA5	Apple_39:19:61	421.78	Low
165	AP (Advantec_f5:cf:a4)	304.0	Low
167	AP (Advantec_f5:d3:a0)	214.66	Low
TC8717TA5	Apple_13:40:f3	592.66	Mid-High
WIFIFC3D5E	AP(HonHaiPr_fc:3d:62)	607.13	Mid-High

To get this Average of Mean value, first traffic per minute was calculated and its mean over a day is calculated. Then an average of all the mean traffic per minute per day day is taken. Average is over 120+ days

Though at this point, the traffic volume of the devices is not as high as what we would observe when capturing the packets from inside the network, still the differences in traffic between the devices at this point would help us in identifying the devices.

Hence, just by passively capturing packets that are in the vicinity without even joining the network and taking the Packet Length information available, an intruder can easily calculate the total traffic associated with every device. With that information he can very well identify IoT devices. Those would be the devices with a significantly lower traffic volume when compared to the rest of the devices as they will be performing only the task they are dedicated for.

5.7.3 Do Devices consistently send the same amount of traffic everyday?

The traffic volume of a device varies with usage. This usage is influenced by the purpose of the device or the user or even the environment. For example, a Smart TV's traffic volume would be directly proportional to the hours of watching TV (influenced by the user) as well the type of content streamed (poor quality, HD, Ultra HD or even mp3 etc.,). A smart light bulb's traffic, similarly, depends on the hours of usage i.e., directly proportional to the amount of time people are present in the vicinity. For any device, measuring the fluctuations in this traffic volume gives us insight into the type of device. Non-IoT devices, in general, have more fluctuations in their traffic volume; because they are used for a number purposes and their traffic volume could vary based on a lot of factors. For example, we saw earlier how the traffic volume of Smart TVs can vary according to the content streamed; it could also very depending on the time and season; people are more likely to watch more TV in the weekends and in the evenings and also during major events like national/world tournaments of popular sports etc., A laptop could be used heavily for a day to browse contents from the Internet, access cloud based applications etc., and not used at all on another day. But IoT devices tend to have less variations in the traffic unless there is a radical change in the environment. Example, the smart light bulb would have more or less similar volumes of traffic unless the occupants are out of the house for a very long period (vacationing for example), making the light bulb to lapse into a sleep mode for longer duration. A Nest Thermostat would always be functioning and collecting data about its surrounding unless the winter is over and it is switched off. Hence, non-IoT devices, because their usages vary largely by purpose, have traffic that is highly fluctuating whereas IoT devices, often dedicated to a single task, have traffic that is quite constant unless there is a radical change in their environment.

Hence, we are going to find out how much the traffic volume of various devices present under different networks vary over a period of the days we collected the data and try to get insights into the type of device they are.

In order to also classify and group the devices based on their rate of traffic fluctuation, we would take the standard deviation values of their mean traffic and assign categories based on the SD. Table 5.18 on page 52 shows the standard deviation ranges of the traffic volume (in Bytes) and the fluctuation categories assigned to them. The lower a Standard Deviation value, the lesser the fluctuation of Traffic Volume over the the days.

Table 5.18: Standard Deviation Limits for Traffic Fluctuation Category

STDEV Range	Fluctuation Category	STDEV Range	Fluctuation Category	STDEV Range	Fluctuation Category
0	Level 0	1001 - 1500	Level 11 I	6001 - 6500	Level 16 I
1 - 100	Level 1	1501 - 2000	Level 11 II	6501 - 7000	Level 16 II
101 - 200	Level 2	2001 - 2500	Level 12 I	7001 - 7500	Level 17 I
201 - 300	Level 3	2501 - 3000	Level 12 II	7501 - 8000	Level 17 II
301 - 400	Level 4	3001 - 3500	Level 13 I	8001 - 8500	Level 18 I
401 - 500	Level 5	3501 - 4000	Level 13 II	8501 - 9000	Level 18 II
501 - 600	Level 6	4001 - 4500	Level 14 I	9001 - 9500	Level 19 I
601 - 700	Level 7	4501 - 5000	Level 14 II	9501 - 10000	Level 19 II
701 - 800	Level 8	5001 - 5500	Level 15 I	10001 and above	Level 20
801 - 900	Level 9	5501 - 6000	Level 15 II		
901 - 1000	Level 10				

Lower Standard Deviation Values have lower Level numbers. Hence, the smaller the Level number, the lesser the fluctuation in Traffic.

Table 5.19 on page 53 shows the traffic fluctuation categorization of devices present under network 'HostNetwork'. Figure 5.6 on page 53 shows the same data in a clear graph. The laptop, device Apple_86:34:26, Samsung TV and Raspberry Pi all have low traffic fluctuations. As we discussed earlier, these devices must be active on the wired interface and use the wireless interface just to send power management information. Hence their traffic as well the fluctuation is low. Nintendo_fa:9d:30 (which is also present under the 'HostNetworkGuest' sometimes) has a higher traffic but very low fluctuation; and from previous section, we know that its traffic is purely data traffic.

Table 5.19: Location A - Network 'HostNetwork' - Fluctuations in Traffic over all days

Device	Avg. of Mean	Standard Deviation	Variance	Fluctuation Category
Laptop (1c:ce:28)	55.93	17.14	395.52	Level 1
Apple_86:34:26	62.4	26.24	903.4	Level 1
Samsung TV	63.43	27.26	1014.32	Level 1
Raspberry Pi	90.18	35.68	1939.18	Level 1
Nintendo (fa:9d:30)	1231.17	104.2	33095.44	Level 2
NestLabs Thermostat (12:05:00)	303.89	199.59	50260.36	Level 2 (Borderline)
NestLabs Thermostat (13:7a:55)	313.47	191.61	47110.96	Level 2
NestLabs Thermostat (14:48:4b)	323.65	187.32	39706.63	Level 2
NestLabs Thermostat (14:58:68)	336.58	191.29	41726.27	Level 2
NestLabs Thermostat (15:3c:6e)	330.45	187.5	40427.83	Level 2
NestLabs Thermostat (2b:2b:61)	326.54	190.12	41903.13	Level 2
Apple_0c:59:e5	171.5	213.34	46197.25	Level 3
Apple_c7:eb:05	569.63	641.25	710998.51	Level 6
Apple_61:b1:70	366.19	699.0	1230844.44	Level 7 (Borderline)
Apple_15:b9:21	627.2	797.93	943990.0	Level 8
Roku AP	918.39	835.4	1025965.0	Level 9
Apple TV	1769.52	840.02	844540.0	Level 9
Apple_0f:fd:22	792.37	1316.39	2991537.5	Level 11 I
Nintendo (09:28:c7)	641.01	1023.49	1100150.0	Level 11 I
Roku (a4:6d:16)	924.62	1049.94	2364625.0	Level 11 I
Apple_de:65:91	4927.26	10607.8	147234146.34	Level 20
AP	17442.26	13325.16	226454032.26	Level 20

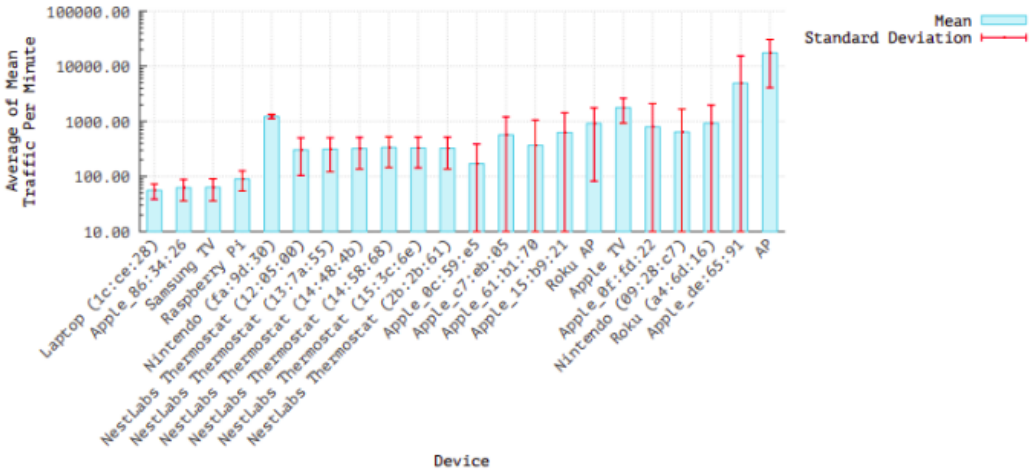

Figure 5.6: Location A - Network 'HostNetwork' - Fluctuations in Traffic

Hence this device was used a lot consistently over the days. All the Nest Thermostat devices have very low traffic fluctuation. This is quite reasonable and in par with

Table 5.20: Location A - Network 'HostNetworkGuest' - Fluctuations in Traffic over all days

Device	Avg. of Mean	Standard Deviation	Variance	Fluctuation Category
Raspberry Pi	80.73	39.14	1705.89	Level 1
Apple_a5:db:34	119.16	60.08	8407.94	Level 1
Nintendo (fa:9d:30)	1231.17	104.2	33095.44	Level 2
Apple_e8:26:91	153.31	134.51	49975.0	Level 2
Apple_5b:81:a1	126.75	110.16	17438.0	Level 2
Apple_4a:aa:13	1032.92	784.83	2775038.46	Level 8
AP	854.78	1772.88	7743016.13	Level 11 II
Apple_3e:39:83	1754.51	4223.03	38156666.67	Level 14 I

our previous observation; they periodically send only power management information to the AP over the wireless interface and there is no remarkable fluctuation in the rate at which they change states, hence not much of a fluctuation in their power management traffic either. Devices Apple_0c:59:e5, Apple_c7:eb:05, Apple_61:b1:70 and Apple_15:b9:21 all have higher levels of traffic fluctuation than the Nest devices, Pi etc., This leads to the conclusion that these are non-IoT devices whose traffic is typical of devices that are used for a variety of purposes that incur a high traffic. Apple TV has a higher traffic fluctuation. Since it sends only power management data, this leads to the conclusion that it stays in Active and Sleep states for varying periods of time and its traffic fluctuates according to the duration of its stay in any one state; the lesser the duration, the higher will be the traffic as it will intimate the AP every time it goes to Sleep mode and comes into Active mode and so will be the vice versa - the longer the duration, the lower will be the traffic as the messages to the AP will now be spaced in time a lot longer. Nintendo_09:28:c7 has lower traffic than the other Nintendo device and its fluctuation is also higher; this means that it was used only a few times with varying degrees of usage. The Roku device has a higher fluctuation than that of the Roku AP and its traffic is also mostly power management information. This could be because the Roku receives its streaming data from the AP over the wired interface. Device Apple_de:65:91 has a very high traffic

and highly fluctuating too. Since we know all of its traffic on the wireless interface is power management data, we confirm that it keeps toggling between the Sleep and Active states over the wireless interface, which results in this high traffic. Also the fluctuation in traffic is consistent with the results from Figure 5.3 which shows that the frequency with which it toggles varies largely. Lastly, we see that the AP has the highest fluctuating traffic among all the devices; also we know that its traffic is more than 90% data traffic. This is reasonable as all the devices active over the wireless interface would communicate via the AP (the router in a home environment) and the traffic volume of the AP varies largely as devices come and leave the network, send and receive different amounts of traffic according to their purpose etc.,

Table 5.20 on page 54 shows the traffic fluctuation categorization of various devices present under network 'HostNetworkGuest'. The Raspberry Pi device has the lowest level of traffic fluctuation; this is due to the fact that it transmits all of its collected data over the wired interface and just sends power management information over the wireless interface. Device Apple_a5:db:34 has very low traffic and fluctuation level. But not much can be said about it as only very minimal data was collected from it. But since the traffic is very low, it is either a non-IoT device that is a little out of the capture range of the Raspberry Pi device or it is an IoT device with very low traffic in general. Device Apple_4a:aa:13 is definitely a non-IoT device as its traffic is quite high and so is its fluctuation level; only devices that are used for various purposes could have this level of traffic volume fluctuation. Devices Apple_e8:26:91 and Apple_5b:81:a1 have fluctuation levels higher than the Nintendo device but lower than the AP. The Access Point has very high levels of traffic fluctuation; because, as said before, it relays the communication between the various devices and its traffic volume is greatly influenced by the number and type of devices present under its network at any instance.

Table 5.21 on page 56 shows the information about various devices present under networks that are foreign to the location of the data capture. Most of the devices are Access Points; but still we that their fluctuation categories vary greatly; this is because these APs are present at different distances from the point of data capture and the data captured from them vary greatly in amount. Hence, this data is not perfectly reflective of the type of devices; it is affected by the amount of data captured from them. But we see that some other Apple and Motorola devices have higher volumes of traffic and higher traffic fluctuation which leads to the conclusion that they must be non-IoT devices. Devices 82:e2:f7:a6:0d:bc and Apple_0e:38:9c have very high levels of traffic and fluctuation; they are definitely non-IoT devices.

Table 5.21: Location A - Other Networks - Fluctuations in Traffic over all days

Network	Device	Avg. of Mean	Standard Deviation	Variance	Fluctuation Category
TC8717TA5	Apple_e1:32:01	63.0	0.0	0.0	Level 0
Unknown Network 4	Motorola_1d:d1:10	686.0	0.0	0.0	Level 0
161	AP (Z-Com_a2:a9:34)	97.55	2.94	24.75	Level 1
TC8717TA5	AP(Technico_07:d7:ab)	107.87	52.2	104967.68	Level 1
166	AP (Advantec_f5:d3:ce)	284.0	31.11	3872.0	Level 1
165	AP (Advantec_f5:cf:a4)	304.0	81.46	19097.6	Level 1
Unknown Network 6	Apple_ea:04:04	526.0	70.78	18228.0	Level 1
Unknown Network 16	AP (SenaoNet_11:c4:ee)	1848.3	102.49	73779.25	Level 2
flynn_Ext	AP (BelkinIn_a4:a2:fe)	162.83	114.73	41378.4	Level 2
167	AP (Advantec_f5:d3:a0)	214.66	129.97	20746.65	Level 2
Hightower Power-1	AP (HonHaiPr_da:c3:73)	286.95	233.75	67532.67	Level 3
Unknown Network 8	6a:31:96:8c:d9:fc	1051.75	254.91	129960.0	Level 3
Unknown Network 15	SamsungE_c8:4a:85	6965.35	383.8	294598.0	Level 4
WIFIFC3D5E	AP (HonHaiPr_fc:3d:62)	607.13	555.54	822900.0	Level 6
TC8717TA5	Apple_39:19:61	421.78	673.02	2720505.26	Level 7
TC8717TA5	Apple_13:40:f3	592.66	710.99	542822.0	Level 8
Unknown Network 6	Apple_a4:11:69	2482.2	700.64	3522120.0	Level 8 (Borderline)
Unknown Network 4	Motorola_fa:89:3e	2337.51	2140.22	48145555.56	Level 12 I
Unknown Network 8	82:e2:f7:a6:0d:bc	4471.72	2466.7	6262140.0	Level 12 I
Unknown Network 6	Apple_0e:38:9c	3462.67	3737.3	24064096.0	Level 13 II

5.7.4 Do Devices only send data or receive data or both ?

Traffic Direction can be of two types - Incoming and Outgoing; Incoming traffic is the traffic that a device receives and Outgoing traffic is a traffic that the device

sends out. Observing the traffic direction that a device gets involved in will us a lot about the type of type and would help us differentiate IoT devices from the non-IoT ones. IoT devices are designed to primarily collect data about their surroundings via the in-built sensors and send the collected data to an outside server that is usually dedicated for receiving the data from the devices of a particular vendor. For this purpose, they use the Internet; to send the data to the outside servers. In return, they do not receive much. The data they sent is processed on the servers for insights about the environment and sometimes, instructions are sent back to the IoT device from the server when some action has to be taken by the device. Hence, they send out so much traffic but in return receive only less traffic. Non-IoT devices will not follow this trend. Depending on what they do, they would either send and receive equal amounts of traffic (not necessarily at the same time) and sometimes even receive more than what they send out. For example, a Laptop would receive more traffic than it is transmitting when a movie is being streamed; but it would be sending more traffic than it is receiving when a huge file is being uploaded to the cloud. A smart TV would almost always be receiving a lot of traffic than it is sending since its primary purpose is to stream media content from the Internet.

Hence just by observing the traffic from outside the network without joining it and by looking at the direction of traffic associated with each device, an intruder can easily differentiate IoT devices from non-IoT ones. Devices with predominantly outgoing traffic would be IoT devices. Devices with more or less equal outgoing and incoming traffic would be laptops, mobile phones or even routers (which can be further identified as ones with very high amounts of incoming and outgoing traffic compared to the rest of the devices in the network). Devices with consistently high incoming traffic would be streaming devices like Smart TVs, Roku etc.,

Table 5.22 on page 58 shows the percentage of days (over all 120+ days) where the traffic is classified based on the direction into 3 different categories - Both In and Out

Table 5.22: Location A - Network 'HostNetwork' - Classification based on Traffic Direction : Both In and Out, Only Out and Only In

Device	Both In and Out(%)	Only Out(%)	Only In(%)	Direction Category
Nest Thermostat (12:05:00)	0.0	100.0	0.0	Exclusively Out
NestLabs Thermostat (13:7a:55)	0.0	100.0	0.0	Exclusively Out
NestLabs Thermostat (14:48:4b)	0.0	100.0	0.0	Exclusively Out
NestLabs Thermostat (14:58:68)	0.0	100.0	0.0	Exclusively Out
NestLabs Thermostat (15:3c:6e)	0.0	100.0	0.0	Exclusively Out
NestLabs Thermostat (2b:2b:61)	0.0	100.0	0.0	Exclusively Out
Apple_0c:59:e5	0.0	100.0	0.0	Exclusively Out
Apple_61:b1:70	0.0	100.0	0.0	Exclusively Out
Apple TV	0.0	100.0	0.0	Exclusively Out
Roku	0.0	100.0	0.0	Exclusively Out
Apple_15:b9:21	33.33	66.67	0.0	Only Out > Mixed
Samsung TV	1.28	98.72	0.0	Only Out $\gg$ Mixed
Apple_c7:eb:05	1.49	98.51	0.0	Only Out $\gg$ Mixed
Nintendo (fa:9d:30)	11.11	88.89	0.0	Only Out $\gg$ Mixed
Apple_0f:fd:22	12.5	87.5	0.0	Only Out $\gg$ Mixed
AP	73.17	26.83	0.0	Mixed $\gg$ Only Out
Laptop (1c:ce:2)	80.0	20.0	0.0	Mixed $\gg$ Only Out
Apple_86:34:26	87.27	12.73	0.0	Mixed $\gg$ Only Out
Raspberry Pi	72.41	27.59	0.0	Mixed $\gg$ Only Out
Nintendo (09:28:c7)	100.0	0.0	0.0	Exclusively Mixed

Table 5.23: Location A - Network 'HostNetworkGuest' - Classification based on Traffic Direction : Both In and Out, Only Out and Only In

Device	Both In and Out(%)	Only Out(%)	Only In(%)	Direction Category
Apple_a5:db:34	0.0	100.0	0.0	Exclusively Out
Apple_e8:26:91	0.0	100.0	0.0	Exclusively Out
Apple_4a:aa:13	0.0	100.0	0.0	Exclusively Out
Apple_3e:39:83	37.33	62.67	0.0	Only Out > Mixed
Apple_5b:81:a1	27.27	72.73	0.0	Only Out $\gg$ Mixed
Nintendo (fa:9d:30)	11.11	88.89	0.0	Only Out $\gg$ Mixed
AP	67.48	32.52	0.0	Mixed > Only Out
Raspberry Pi	100.0	0.0	0.0	Exclusively Mixed

(when a device participates in sending as well as receiving traffic over the entire day), Only Out (when a device only sends the data over an entire day) and Only In (when a device only receives data over an entire day). We see that the Nest Thermostat devices only Send Out the data. They do not receive anything on the wireless interface; they

send out the power management information. To get the information about the true traffic of the Nest Devices, we should look into their traffic over the wired interface. Other devices that are expected to receive more traffic than they send out like Apple TV, Samsung TV, Roku etc., also have either exclusively outgoing traffic or the days where they have mixed traffic is negligible when compared to the days when they have exclusively Out traffic; this could be the result of the devices being using the wired interface for all their streaming activities (which would reflect their true traffic direction) and using the wireless interface for just sending power management information. Of the two Nintendo devices, device Nintendo_fa:9d:30 has exclusively outgoing traffic for most of the days whereas Nintendo_09:28:c7 sends and receives traffic on all the days that the data was captured. From previous sections, we know that this traffic is purely data traffic. But as to why one device behaves differently from the other, we have to know exactly what is being sent by Nintendo_fa:9d:30; we would get that information only when we observe the traffic from inside the network. The AP as expected, spends most of the days sending and receiving the traffic; it would be relaying data between the devices to enable them to communicate. The Raspberry Pi device has predominantly mixed traffic and on some days exclusively Outgoing traffic. The outgoing traffic is definitely the power management information on the wireless interface.

Table 5.23 on page 58 shows the traffic direction information for the devices present under network 'HostNetworkGuest'. Devices Apple_a5:db:34, Apple_e8:26:91 and Apple_4a:aa:13 all have exclusively outgoing traffic; device Apple_3e:39:83 has mostly outgoing traffic with some days having mixed traffic. We know that their traffic is purely power management information and no data. Either the devices are inactive over the wireless interface or they do not do anything else at all. The Raspberry Pi device sends as well as receives traffic on all days; from previous analysis, this traffic is exclusively type 36 packets.

Table 5.24 on page 60 shows the data about the traffic direction of the various devices present under different networks in the vicinity. We see that all of the APs except Technico_07:d7:ab have exclusively outgoing traffic. We saw earlier that most of the time, their traffic was purely data (Subtype 32). The absence of incoming traffic could be because the devices under these APs could be so far away from the point of observation that we were not able to capture them; but the AP's transmission range boundary could overlap with our range of data capture that we were able to capture the packets transmitted from those APs. AP Technico_07:d7:ab is the only exception as we have managed to captured both incoming as well as outgoing traffic from it; so it quite nearer compared to other APs. That is why we are able to see more number of devices from network 'TC8717TA5' (to which the AP belongs to) than the other networks.

Table 5.24: Location A - Other Networks - Classification based on Traffic Direction : Both In and Out, Only Out and Only In

Network	Device	Both In and Out(%)	Only Out(%)	Only In(%)	Direction Category
TC8717TA5	Apple_e1:32:01	0.0	100.0	0.0	Exclusively Out
Unknown Network 4	Motorola_1d:d1:10	0.0	100.0	0.0	Exclusively Out
161	AP (Z-Com_a2:a9:34)	0.0	100.0	0.0	Exclusively Out
166	AP (Advantec_f5:d3:ce)	0.0	100.0	0.0	Exclusively Out
165	AP (Advantec_f5:cf:a4)	0.0	100.0	0.0	Exclusively Out
Unknown Network 6	Apple_ea:04:04	0.0	100.0	0.0	Exclusively Out
Unknown Network 16	AP (SenaoNet_11:c4:ee)	0.0	100.0	0.0	Exclusively Out
flynn_Ext	AP (BelkinIn_a4:a2:fe)	0.0	100.0	0.0	Exclusively Out
167	AP (Advantec_f5:d3:a0)	0.0	100.0	0.0	Exclusively Out
Hightower Power-1	AP (HonHaiPr_da:c3:73)	0.0	100.0	0.0	Exclusively Out
Unknown Network 8	6a:31:96:8c:d9:fc	0.0	100.0	0.0	Exclusively Out
Unknown Network 15	SamsungE_c8:4a:85	0.0	100.0	0.0	Exclusively Out
WIFIFC3D5E	AP (HonHaiPr_fc:3d:62)	0.0	100.0	0.0	Exclusively Out
TC8717TA5	Apple_39:19:61	0.0	100.0	0.0	Exclusively Out
TC8717TA5	Apple_13:40:f3	0.0	100.0	0.0	Exclusively Out
Unknown Network 6	Apple_a4:11:69	0.0	100.0	0.0	Exclusively Out
Unknown Network 4	Motorola_fa:89:3e	0.0	100.0	0.0	Exclusively Out
Unknown Network 8	82:e2:f7:a6:0d:bc	0.0	100.0	0.0	Exclusively Out
Unknown Network 6	Apple_0e:38:9c	0.0	100.0	0.0	Exclusively Out
TC8717TA5	AP (Technico_07:d7:ab)	50.98	46.08	2.94	Mixed ≈ Only Out

5.8 Conclusion

Hence, by just observing the packets from outside the network passively and examining the traffic direction of devices, an intruder can guess the type of device simply by resolving the OUI present in the first half of the MAC address. This would narrow down the list of what type of device it could possibly be. And from the Beacon frames he can get the list of nearby network names and their Access Point addresses. From the data packets, analyzing the 4 different address fields and the DS flag under Frame Control would give him the list of APs and the devices under it. Combining both the lists, he can figure out which devices belong to what network. And from the signal strength information, he can figure out which network belongs to which house. Hence effectively, just from this superficial information, one can figure out exactly the number and to a greater degree, the type of devices that are present in a house. And from further analyzing the data, he can clearly differentiate an IoT device from a non-IoT device by looking at how active a device is and how often does it go to sleep. IoT devices would not be as active as a non-IoT device and they would stay in sleep state for a longer duration consistently so as to save battery. This is verified by the results we got from our analysis. By analyzing the traffic volume associated with a device and the various other parameters like the what type of packets they send, their traffic volume, how consistently do they send the same volume of traffic, whether they only send or receive packets or both and to what percentage - all this would give him sufficiently clear results to differentiate the various devices present inside the network. By combining results of all the parameters for each device, he would be able to create a profile for each device. Since he would already know the type of device from the OUI data, he can associate that profile with the exact kind of device. This composed profile would be a good way to identify similar devices (or even exactly the device) when presented with a new set of devices.

CHAPTER 6

ANALYSIS OF DATA CAPTURED FROM INSIDE THE NETWORK

The previous chapter dealt with what an intruder could learn just by passively observing the traffic of the network from a distance without even joining the network. Though we were able to learn about how IoT and non-IoT devices differ in their behaviour under various parameters, we were not able to learn exactly what activity these devices are involved in. But now, since we are going to observe the traffic from inside the network, it would give us a much more detailed insight about the activities of the devices.

From this internal traffic, we would try to identify IoT devices and differentiate how they vary in their behaviour from conventional non-IoT devices. Since we are inside the network, we would be able to see almost all the traffic that is exchanged within the network (sometimes we may lose some packets if the network is too large for the capturing device to cover) and that would give us much more information to go on about our analysis than the information we got from observing the traffic from outside. Hence, before classifying these devices on the parameters that we already employed in the previous section like Traffic Volume, Traffic Direction etc., we would try to analyze the information that is exclusively available only at this point, i.e., from inside the network.

6.1 Data Encapsulation

In Table 5.1 on page 18, we saw the fields present in IEEE 802.11 header, which is the encapsulation employed when a packet leaves the AP. Inside the network, IEEE

802.3(Ethernet) header is used. Generally, all wireless packets use 802.11 header and all wireless nodes send Data, Management and Control frames. But traditional OSes do not know how to interpret this header; they can only interpret 802.3 header. So the NIC converts the 802.11 header into 802.3 header before passing it up to the OS. Hence , the OS always sees only the 802.3 header.

Table 6.1 on page 63 shows a subset of the various fields present in IEEE 802.3 header. This is the header that visible at the OS for all devices present in wired as well as wireless medium.

Table 6.1: A subset of fields present in the IEEE 802.3 Header

Layer	Field Name	Description
Frame *	frame.time	Timestamp of the packet capture
	frame.interface_name	The interface on which the packet was captured
	frame.length	Packet Length
	frame.time_delta	Time elapsed after the capture of the previous packet
	frame.protocols	Protocols present in the packet
Ethernet	eth.type	Type of Ethernet frame; value varies depending on the higher layer protocol
	eth.src	Source MAC address
	eth.dst	Destination MAC address
	eth.ig	Individual or Group address; 0 indicates unicast, 1 indicates multicast or broadcast address
	eth.lg	Local or Global address; 0 indicates Vendor given, 1 indicates local; Generally vendor given addresses are not changed but may be changed for administrative purposes
Network	ip.src	Source IP address
	ip.dst	Destination IP address
	ip.id	Identification; for identifying the frames in case there is fragmentation
	ip.ttl	Time to Live
	ip.geoip.src_asnum	Source Autonomous System number
	ip.geoip.dst_asnum	Destination Autonomous System number
Transport	tcp.srcport	TCP Source Port
	tcp.dstport	TCP Destination Port
	tcp.flags.syn	TCP SYN Flag
	tcp.flags.ack	TCP ACK Flag
	tcp.flags.fin	TCP FIN Flag
	udp.srcport	UDP Source Port
	udp.dstport	UDP Destination Port

Frame fields are captured information provided by tshark and is not an OSI Layer

Using the information obtained from this header, we would like to learn about the activities of each device and gain meaningful insights about the devices. This information (except the MAC address) is available only inside the network and is not visible from outside the network.

6.2 Network Protocols used by a Device

Network Protocols can be defined as rules that are practised by all the devices present in the network in order to facilitate communication with each other as well as to transport packets from the source to their intended destination. Different protocols are defined on different layers of the OSI model to carry out the various functions defined for that layer. For example, TCP (Transmission Control Protocol) is the most commonly used protocol on the Transport Layer for reliable communication; FTP (File Transfer Protocol) is used for transferring files across clients in the network and it acts on the Application Layer; ARP (Address Resolution Protocol) is used for resolving IP addresses into MAC addresses when devices in LAN want to communicate and it acts on the Media Access Control Layer. These are a few examples of the various protocols operating in different layers on the OSI model.

Hence by learning what protocols are used by various devices, we would be able to say what exactly these devices are doing. Also, the number of protocols employed by each device must also differentiate IoT and non-IoT devices. As discussed before, since IoT devices perform a single or a very few closely related tasks, the number of protocols used by them must be much lower than the non-IoT devices like laptops, smart phones etc., since they perform a variety of tasks and would be using a variety of protocols to fulfill them.

Since devices are connected to the network either via the wired interface or the wireless interface or some times both, we would analyze the protocols used by the devices on both the media.

6.2.1 Wired Medium

Table 6.2: Location A - Network 'HostNetwork' - Protocols used by devices on Wired Medium while Transmitting

Device	Protocols Used	Count
Egauge (00:00:b8)	igmp,ip,	2
ObihaiTe (62:e9:8e)	arp,ip,	2
Egauge (00:04:a4)	igmp,ip,	2
Laptop (47:d9:d9)	arp,ip,llc,	3
MacMini (ba:b3:d6)	arp,ip,ipv6,	3
Apple_ea:b3:5e	llc,vssmonitoring,wlccp,	3
Nintendo (09:28:c7)	arp,bootp,llc,	3
Nintendo (fa:9d:30)	arp,bootp,ip,llc,	4
Azurewav_37:19:02	arp,ip,ipv6,llc,	4
Insteon Hub (46:fa:a7)	arp,bootp,data,ip,	4
Raspberry Pi 2 (28:4c:14)	bootp,icmpv6,igmp,llc,mdns,	5
Washing Machine (34:2d:38)	arp,bootp,ip,llc,ssdp,	5
Dryer (34:3d:61)	arp,bootp,ip,llc,ssdp,	5
Raspberry Pi 1 (35:bb:d5)	bootp,icmpv6,igmp,llc,mdns,vlan,	6
Roku (72:1a:5d)	arp,bootp,data,igmp,ip,llc,ssdp,	7
NestLabs (12:05:00)	arp,bootp,dhcpv6,icmpv6,igmp,ip,ipv6,llc,mdns,	9
NestLabs (13:7a:55)	arp,bootp,dhcpv6,icmpv6,igmp,ip,ipv6,llc,mdns,	9
NestLabs (14:48:4b)	arp,bootp,dhcpv6,icmpv6,igmp,ip,ipv6,llc,mdns,	9
NestLabs (14:58:68)	arp,bootp,dhcpv6,icmpv6,igmp,ip,ipv6,llc,mdns,	9
Tp-LinkT_49:07:24	arp,ath,bootp,data,igmp,ip,ipv6,kink,llc,manolito,pn_io,	11
Apple TV (50:6a:f1)	arp,bootp,data,dhcpv6,icmpv6,igmp,ip,ipv6,llc,mdns,tcp,wlccp,	12
Mac Mini (aa:8d:ef)	arp,bootp,dhcpv6,icmpv6,igmp,ip,ipv6,llc,mdns,nbdgm,nbns,ssdp,	12
Raspberry Pi 1 (b7:e0:1f)	arp,bootp,dns,http,icmpv6,igmp,ip,ipv6,ntp,Protocol,ssh,tcp,	12
Laptop (1c:ce:28)	arp,bootp,data,db-lsp-disc,igmp,ip,llc,mdns,nbdgm, nbns,ssdp,vssmonitoring,wlccp	13
Printer (08:bf:14)	arp,bootp,data,dhcpv6,icmpv6,igmp,ip,ipv6,llc, llmnr,mdns,nbns,srvloc	13
Apple_de:65:91	arp,bootp,data,dhcpv6,esp,icmpv6,igmp,ip,ipv6,llc, mdns,ssdp,vssmonitoring,wlccp	14
Router (e5:a8:02)	afp,arp,bootp,data,dns,fb_zero, gryphon,http,icmpv6,ip,ipv6,isakmp,mdns, nbdgm,ntp,portcontrol,ssh,ssl,tcp,wol	20
Samsung TV (e1:c2:0a)	arp,ath,bootp,data,dhcpv6,elasticsearch, enip,hcrt,icmpv6,igmp,ip,ipv6,kink, llc,manolito,pn_io,ssdp,tc_nv,tcp,tzsp	20

Table 6.2 on page 65 shows the protocols used by various devices on the wired medium while transmitting. At the outset, we can clearly see the difference in the number of protocols employed by various devices. Devices like Egauge, Insteon Hub, washing machine and dryer, Nest Thermostat devices employ a lesser number of protocols when compared to devices like Samsung TV, Router, Laptop, Apple TV etc., This clearly illustrates the difference in the nature of the two types of devices. IoT

devices, since they perform a minimum number of tasks employ a minimum numbers of protocols to do just that whereas devices like Laptop, Routers, Smart TVs etc., are used to perform a much wider array of tasks and the variety of protocols employed by them reflect that.

Egauge devices consistently use protocol IGMP. IGMP (Internet Group Management Protocol) is used by hosts and multicast-enabled routers to perform multicast communication. Multicast is different from broadcast in that a broadcast packet reaches every host in the LAN whereas a multicast packet reaches only hosts present in the multicast group. In order to maintain the group, multicast routers send general and group specific queries to maintain the group status and reception status of hosts respectively. Hosts wishing to remain in the multicast group must reply to this as a 'Membership Report'. Egauge is doing that by periodically sending membership report packets to remain in the group. Here it reports to be present in group '224.0.0.251' which is the multicast IP of MDNS. Device Egauge_00:27:cc is involved in ARP, IGMP, and MDNS. ARP (Address Resolution Protocol) is used when a device wants to know the MAC address of the device which has a particular IP. Here it asks for MAC address of 10.0.1.1 which is the default gateway in many home network settings. It also sent MDNS packets. MDNS (Multicast DNS) is used for resolving host addresses when a local name server is absent (which is usually the router). The host sends an MDNS query asking the host with the domain name to identify itself; the host concerned replies back to the same multicast IP with its IP in the reply as its source. This reply is received by all in the multicast group and they update their DNS cache.

We see all the Nest Thermostats employing the same protocols indicating the uniformity in their behaviour. Periodically they sent ARP requests asking for 10.0.1.1 which is the default gateway. Bootp (Bootstrap) is a protocol that is used to assign IP addresses automatically to devices that come online. The difference between Bootp

and DHCP (Dynamic Host configuration protocol) is that bootp assigns a static address and DHCP assigns an address dynamically from the pool of IP addresses that are currently available. A bootp message is sent by a client to the server when it boots up and needs an IP address. Otherwise bootp is not used. But DHCP is used for on-demand IP address and it is compatible with bootp as well. Hence, generally bootp is used in immobile devices. Therefore the presence of bootp indicates not only that the device is stationary but the timestamp indicates the time when it was booted as well. Of the 40 days' data we used for analysis, Nest Devices used bootp only exactly on 3 days - 15/Mar/2018, 31/May/2018 and 21/Jun/2018. These could very well be the times these devices were booted or restarted. We also see that bootp is used in devices like Samsung TV (which has sent the message almost daily - which is very reasonable for a TV which would be switched off every time after use), Router (e5:a8:02), Printer (08:bf:14) (which has sent the message only on 3 days - might be connected online all the time to facilitate printing whenever needed), Mac Mini, Apple TV (which also sent the message almost daily like Samsung TV), Washing Machine (which sent the message only on 2 days), Dryer (sent bootp only on 1 day) which are all stationary. But we do not see bootp messages in devices like Laptop (47:d9:d9), Apple_ea:b3:5e, Apple_0c:59:e5 etc., which are clearly mobile devices. But there are some exceptions here: ObihaiTe (62:e9:8e), Mac Mini (ba:b3:d6) etc., are clearly immobile. But we do not see any bootp messages associated with them. It could be that in the time period of our observation, they did not boot up. Nest Devices also employ ICMP (Internet Control Message Protocol) that is used for maintaining the proper communication by enabling sending of error messages, facilitating managing of multicast groups. They regularly send ICMPv6 (version 6) Router Solicitation messages (Type 133), which are sent with an unspecified address (::) as the source IP address till they get a valid IP address and once they get an IP, they send a Multicast Listener Report (Type 143) Message with the assigned IP as the source.

The number of ICMP messages vary greatly over the data collection period. During the months of January and February, the number of ICMP messages are quite high and they reduce gradually over the coming months. In the month of June, there is an average of around 10 ICMP messages per day that shows reduced activity. This is reasonable as the Thermostat would be quite active during winter months and not so active during summer.

TCP (Transmission Control Protocol) is used to send data in a reliable manner as acknowledgements are received for every packet that is sent. It is the most commonly used protocol for sending actual data across the network. UDP is generally used when the data sent is short bursts of message that do not have to be tracked for reliability. We see that TCP is employed by devices like Router (e5:a8:02), Raspberry Pi 1 (b7:e0:1f), Samsung TV, Apple TV etc.,; Router and Raspberry Pi sending the data that is being collected in the network to the remote server and both the Smart TVs streaming content from the Internet. WLCCP (Wireless LAN Context Control Protocol) is used in an environment where there are multiple APs to provide services commonly called as WDS (Wireless Domain Services); WBS provides enhanced security, scalable WLAN etc., This is implemented in Cisco Access Points. We see it in Apple_ea:b3:5e (which we clearly know to be an AP from previous sections) but not in Router (e5:a8:02), which could mean that Apple_ea:b3:5e is a Cisco AP. We see the Printer (08:bf:14) uses protocol srvloc (Service Location). It is a service discovery protocol used by devices to announce the services they provide over a local network when they act as the Service Agents(SA). Devices wanting to use the service act as User Agents (UA) and would always be listening on port 427. This protocol is typically used by devices like printers that are required to work on the network without any prior configuration.

Simple Service Discovery Protocol (SSDP) is used by devices to advertise their presence and their services. Devices that want to use the service discover the services

using SSDP. It operates on port 1900 and is the basis of the Plug-and-Play protocol that is commonly used. It is used in small networks and home environments without the need for any configuration. Advertising devices use NOTIFY messages and devices searching for services use M-SEARCH messages. Washing Machine and Dryer send SSDP NOTIFY messages once or twice almost daily. Roku_72:1a:5d sends hundreds of SSDP NOTIFY packets each day. Devices like Laptop (1c:ce:28), MacMini (aa:8d:ef), and Apple_0f:fd:22 send M-SEARCH messages quite frequently throughout the day on many days. The M-SEARCH message from these devices and the NOTIFY message from Roku_72:1a:5d happen back and forth that it could only mean these devices are streaming content from Roku.

NBDGM (NetBIOS Datagram Service) and NBNS (NetBIOS Name Service) are used by some legacy systems (or Windows systems for name resolution when a DNS is absent) that use NetBIOS API for file sharing and connecting with the printer and enable them to communicate over the TCP/IP stack. NBNS acts over UDP port 137 and is used for Name Services like registering a name, lookup for NetBIOS name resolution (as each device has a NetBIOS name) etc., and NBDS acts over UDP port 138 and is used for sending connectionless datagrams (meaning no previous communication is necessary for sending packets) to other NetBIOS devices. From our data, we see that these protocols are used by Printer (08:bf:14), MacMini (aa:8d:ef), Laptop (1c:ce:28), Apple_86:34:26 and Router (e5:a8:02), probably for communicating with the printer as we see NBDS (Datagram packets) originating from all the devices except the Printer, meaning the Printer just receives content from all other NetBIOS devices and just participates in NBNS (Name Resolution). We were also able to learn their NetBIOS names - Router (e5:a8:02) : JEANNIES-AIRPOR, MacMini (aa:8d:ef) : NetBIOs name MACMINI-BEE177, Laptop (1c:ce:28) : MACBOOKPRO-CE28, Apple_86:34:26 : RUGGER9.

6.2.2 Wireless Medium

Table 6.3: Location A - Protocols used by devices on Wireless Medium while Transmitting

Device	Protocols Used		Device	Protocols Used	
	Protocols	Count		Protocols	Count
AP	eapol	1	Washer	arp, bootp, ssdp	3
Egauge_b8	igmp	1	Apple_0f:fd:22	arp, bootp, igmp, mdns	4
Egauge_a4)	igmp	1	Apple_15:b9:21	arp, bootp, igmp, mdns	4
Mosberge_0a	igmp	1	Ipad_38)	arp, bootp, igmp, mdns	4
Nintendo_09	arp	1	Raspberry Pi_b7	arp, bootp, igmp, mdns	4
Ipad_0d	arp, igmp	2	Ipad_3e)	arp, bootp, igmp, mdns, raknet	5
Raspberry Pi_43	data, igmp	2	Apple TV	arp, bootp, icmpv6, igmp, mdns	5
Nest Thermostat_12	arp, bootp	2	Apple_61:b1:70	arp, bootp, data, igmp, mdns	5
Nest Thermostat_13	arp, bootp	2	Apple_de:65:91	arp, bootp, icmpv6, igmp, mdns	5
Nest Thermostat_14:48	arp, bootp	2	Printer	arp, bootp, icmpv6, mdns, nbns	5
Nest Thermostat_14:58	arp, bootp	2	Laptop	arp, bootp, db-lsp-disc igmp, mdns, nbns, smb	7
Nest Thermostat_15	arp, bootp	2	Apple_86:34:26	arp, bootp, icmpv6, igmp, mdns, nbns, smb	7
Nest Thermostat_2b	arp, bootp	2	MacMini_aa	arp, bootp, icmpv6, igmp, mdns, nbns, smb	7
Nintendo_fa	arp, bootp	2	Raspberry Pi_28	arp, bootp, eapol, icmpv6, igmp, ssh, tcp	7
Roku_72	arp, bootp	2	Tp-LinkT	arp, ath, bootp, data, igmp, kink, manolito, pn_io	8
Dryer	arp, ssdp	2	Router	arp, bootp, data, icmpv6, mdns, portcontrol, smb, ssh, tcp, wol	10
Egauge_27	arp, igmp, mdns	3	Samsung TV	arp, ath, bootp, data, elasticsearch, hcrt, igmp, kink, manolito, pn_io, tc_nv	11
Insteon Hub	arp, bootp, data	3			

Table 6.3 on page 70 shows the protocols used by various devices connected on the wireless medium. The number of protocols used by various devices clearly differentiate IoT and non-IoT devices as before.

Over all, we can see that almost all of the devices employ lesser number of protocols over the wireless medium than the wired medium. This indicates that they are more

active over the wired interface. But regarding Nest Thermostats, we do not know if that is case; the data collected on internal wireless medium is during the months of June and July where as the data collected on internal wired stars from January. Since it is winter during January and summer during June/July, the reduced activity of Nest Thermostats could simply be because of the period of the data capture on internal wireless. We see that both the Ipad devices Ipad (41:a9:38) and Ipad (41:a9:3e) use protocol bootp. This is strange because, on the wired medium, only devices that remained stationary were sending bootp messages. Even if we assume that bootp messages are sent whenever the Ipad was bootup, the absence of dhcp is puzzling. Because, in wired medium, even if latest devices like Samsung TV, Apple TV, Mac Mini etc., did send bootp messages when they booted up, they also used dhcp which is the protocol the more advanced devices use for procuring IP addresses. Analyzing the timestamps when bootp was sent, we see that device Ipad (41:a9:38) booted up more frequently (every 2-3 days, sometimes even frequently) than Ipad (41:a9:3e) (mostly only once in a week or 10 days). This could either indicate that device Ipad (41:a9:38) was used more frequently and hence switched off more frequently in need of a recharge or that the two Ipads belong to 2 different people in the household and one charges the device more diligently than the other. Also, one of the Ipad devices, Ipad (41:a9:3e) uses protocol raknet on 3 days continuously 14th, 15th and 16th June, 2018. Raknet is a networking protocol developed by Oculus VR, Inc. On the first day, raknet packets are transmitted for just 5 mins in the morning; on second day for 1 hour in the evening and on the third day for 20 mins in the morning. Hence, on those 3 days, the Ipad device has connected with an Oculus device for the durations mentioned. The rest of the devices employ only a subset of the protocols that they used on the wired medium; this indicates reduced variety of activity over the wireless medium.

Hence, the number of protocols observed on the wireless medium, just like the wired medium, clearly illustrates the difference between IoT and non-IoT devices. IoT devices employ fewer number of protocols reflecting their narrow scope of activities whereas non-IoT devices like Smart TVs, Router, TP Link, Mac Mini etc., use a larger number of protocols indicating the wide variety of tasks they perform.

Thus by merely looking at the number of protocols employed a device, we can clearly differentiate between IoT and non-IoT devices. IoT devices would be using fewer protocols where as non-IoT devices, since they participate in a variety of tasks, employ a larger number of protocols. Also, by examining what type of protocol is used by each device, we can learn a lot about the characteristics of each device including the type of activity each device gets involved in. Egauge devices participate in multicast messaging. About the Nest Thermostats, we know that they participated in multicast messaging groups, we know the exact timestamp of their boot up and that they are stationary (by the use of bootp), and that their degree of activity has reduced after the winter months of January and February. Raspberry Pi, Router and Smart TVs are involved in content streaming which is verified by the presence of TCP. Also, we have confirmed the presence of multiple APs in this single network by the presence of protocol WLCCP which is typically used in a multiple AP network. We know when devices like Laptop (1c:ce:28), MacMini (aa:8d:ef) etc., get connected with Roku for streaming multimedia by observing SSDP packets. We also know that the host network still has some devices that use the NetBIOS over TCP/IP for communicating with other NetBIOS devices. NetBIOS is considered a high security risk as it is vulnerable to 'pass the hash' attack. Thus we were able to learn a lot of information about each of the devices in the network by observing and analyzing the protocols they use.

6.3 Network Ports used for communication

Network ports can be seen as the starting and ending points for network communication. Any data that has to be sent originates from them, goes through all layers of the Network on both the transmitter and receiver side and finally reaches the port on the other end. On a system level, a port is the point or the logical identification of the various processes that are participating in network communication. Some port numbers are assigned for commonly used Internet tasks and they do not change. For example, SSH always acts on port 22, HTTP always acts on port 80, DNS always acts on port 53. Hence just by looking at the port numbers associated with a device, one would be able to say what type of activities the device is involved in. Hence, this would supplement the protocol data that we got from the previous section.

The ports can either act over TCP or UDP. TCP is used when the communication is premeditated by both the sender and the receiver and requires that it has to be reliable ie., loss of packets is not tolerable and packets have to be retransmitted if lost. UDP is used when the sender wants to send traffic just like that to the receiver without acknowledging the receiver first. It is used in tasks where a little loss of packets is quite tolerable; lost packets are not retransmitted. For example, TCP is used in sending email via SMTP; no part of the data must be lost; hence if a packet gets lost, it is retransmitted to ensure all content is delivered. UDP, for example, is often used in video streaming; no problem even if one or two packets are lost as the frames will keep coming and the loss will not be noticeable; here retransmission of packets will create unnecessary overhead. Hence analyzing how much of the traffic sent by a device is TCP/UDP could also help us in understanding what the devices do the devices better.

Table 6.4 on page 74 shows the various ports employed by the non-IoT devices like Laptops, Smart TVs, Ipads etc., during transmission on the wired and the wireless medium.

Table 6.4: Location A - Network Ports Used by Non-IoT Devices while Transmitting

Device	Wired Medium		Wireless Medium	
	Source Port	**Percentage**	**Source Port**	**Percentage**
Apple_0f:fd:22	68	0.73%	68	0.84%
	5353	86.98%	5353	99.15%
	Custom	12.29%		
Apple_c7:eb:05	68	68.06%	68	100%
	4500	31.41%		
	Custom	0.52%		
Apple_de:65:91	68	0.55%	68	0.79%
	4500	0.0029%	5353	99.20%
	5353	94.81%		
	Custom	4.64%		
Ipad (41:a9:3e)	68	0.065%	68	0.29%
	5353	23.76%	5353	38.25%
	Custom	76.18%	Custom	61.46%
Laptop (1c:ce:28)	138	0.023%	138	0.032%
	68	0.08%	68	0.096%
	137	3.44%	137	4.56%
	5353	4.52%	5353	11.57%
	Custom	91.93%	Custom	83.72%
Apple TV (50:6a:f1)	68	0.036%	68	0.0501%
	5353	99.96%	5353	99.95%
	7000	0.0004%		
	Custom	0.0096%		
MacMini (aa:8d:ef)	68	0.94%	68	1.33%
	137	39.76%	137	55.84%
	138	0.038%	138	0.092%
	5353	16.76%	5353	42.74%
	Custom	42.49%		
Samsung TV (e1:c2:0a)	68	0.022%	68	0.12%
	3942	0.18%	Custom	99.88%
	3955	7.81%		
	7678	0.001%		
	8001	53.94%		
Router (e5:a8:02)	22	99.50%	22	98.061%
	53	1.28%	67	0.038%
	67	0.003%	138	0.095%
	80	0.0039%	5351	0.00103%
	138	0.016%	5353	1.80516%
	443	0.011%		
	548	3.86%		
	4500	1.28%		
	5223	0.001%		
	5228	0.0005%		
	5351	0.0003%		
	5353	0.37%		
	8082	0.0005%		
	9543	0.078%		
Printer (08:bf:14)	68	0.23%	68	0.219%
	137	0.701%	137	0.622%
	427	0.35%	5353	99.158%
	3702	0.233645%		
	5353	98.1308%		
	Custom	0.350%		

6.3.1 Wired Medium

The ports used by the devices on the wired medium is reflective of the protocols used by them which we saw in the previous section; devices differ in the number of ports they use and their percentage. Let us take device Apple_0f:fd:22. It uses port 68 for transmission 0.73% of the time. Port 68 is the one used by bootp clients when they bootup and want to send bootp packets to the server. The low percentage of bootp indicates that it did not boot up often. This is verified by the fact that the protcol was used by Apple_0f:fd:22 only on 5 days during the period of our data capture. Hence, from the previous section, we knew that it sent bootp messages. This section gives us quantitative information about it. Port 5353 is used for sending multicast DNS messages. Multicast DNS is used when a true DNS cannot be implemented; for example, small networks implement MDNS where each device functions as the MDNS client as well the server, periodically multicasting their MDNS name, IP address, the services they offer etc., Hence it is common for networks with devices participating in MDNS to have high MDNS traffic. Apple_0f:fd:22 uses port 5353 for 86.98% of the time. This being an Apple device, it is not purely MDNS traffic; the traffic is created by 'Bonjour', Apple's application meant for zero configuration networking. It is aimed at providing services like hostname resolution, address assignment and service discovery etc., seamlessly with zero input from the user and it is implemented over MDNS; that is why the high MDNS traffic. Apple_0f:fd:22 also uses some 12.29% of custom ports. Custom ports are the ones that are out of the 0-1023 range which are reserved for commonly employed tasks. For example, the applications running in our laptops would use custom ports so as to not clash with the standard network applications that would be running under the hood. The presence of custom ports indicate that Apple_0f:fd:22 is running other custom applications.

Device Apple_c7:eb:05 has used port 68 around 68% of the time, which is quite high. From the data, we see that it has booted up almost daily during the period

of our data collection; may be this is a device that is often restarted like a laptop. Another reason this device might be a laptop is it uses port 4500, which is reserved for IPSec NAT Traversal. From the protocol data, we also see that it uses protocol 'esp' - Encapsulating Security Payload. This is used to keep the data secure by encrypting everything from Transport Layer and above. Generally routers would employ a technique called NAPT (Network Address Port Translation) to be able to use a single routable IP address for all the devices inside the network. But this requires a port number to uniquely identify each device. Since ESP encrypts everything from the Transport Layer, a port is not available for NAPT. This is where NAT Traversal comes in. It encapsulates the already encapsulated ESP packet with a UDP header with source and destination ports as 4500 and now NAPT can function properly. Since NAT-T is used, this is not VPN but just Transport Mode of IPSec. This encryption is something only a laptop or a device with a higher resource would do.

Laptop (1c:ce:28) is involved in custom ports for more than 91.93%, meaning that it is running a lot of custom applications (non-system applications). This proves that it could only be a device like laptop or smart phone that is running various applications. It uses port 138 which is reserved for NetBIOS Datagram service. This is in accordance with our observation from the previous section and it is probably used to send data to the printer. Apple_de:65:91 is also involved in IPSec NAT Traversal (port 4500) for a smaller percentage. Around 4.6% of its traffic is from custom applications. The presence of traffic from port 5353 (Bonjour Application) and 4500 (IPSec NAT-T) confirm that this is a non-IoT device and that too, an Apple device. Apple TV's traffic is also mostly from port 5353, which is from Bonjour acting over MDNS. The presence of port 7000 indicates that Apple's Quick Time Streaming Server is running on this machine[36]. It is used for delivering media content like audio and video over the Internet. Therefore, the presence of ports 5353 and 7000 indicate that Apple TV (50:6a:f1) is an Apple device and port 7000 clearly indicates that it

must be a streaming device like TV. MacMini (aa:8d:ef), being an Apple device uses port 5353 for around 16% of the time. It also uses a high amount of custom ports - 42.49%. This indicates that it is running a lot of non-system applications and hence it must be a device like laptop or desktop. It is also involved in NetBIOS operations using ports 137 and 138, which are probably used for maintaining communication with the printer and sending the content to be printed, respectively.

Apple TV (50:6a:f1)'s major traffic is from port 5353, around 99%. This is from the application Bonjour that is running over MDNS. Presence of port 7000 indicates that the device is running Apple's Quick Time Streaming Server. It is used for streaming media content like audio and video over the Internet. Hence, the presence of ports 5353 and 7000 indicate that not only is it an Apple device but also a streaming device like TV. MacMini (aa:8d:ef) is involved in a lot of custom ports, around 42.49%. This means that it is running a lot of non-system applications. This could only mean that this is a non-IoT device running various applications. Being an Apple device, it also uses port 5353 (Bonjour over MDNS) for around 16.76% of the time. It is also involved in NetBIOS traffic on ports 137 and 138 which are used for Naming Service and Datagram Service respectively. This is most probably used for maintaining communication with the printer and sending contents for printing (which is why its percentage is far lesser).

Samsung TV (e1:c2:0a)'s traffic largely is due to port 8001 (53.9%). This port always uses multicast IP 224.0.0.7. From those information, this is most probably running a Streaming Server because multicast 224.0.0.7 is reserved for streaming routers (ST Routers). This is quite expected as the primary purpose of the Samsung TV is to stream content. But this can be a streaming router if it is streaming content for other devices and sending it to them. Its next largest traffic comes from port 3955 (7.81%). This port is used in Simple Service Discovery Protocol (SSDP). The device is sending SSDP NOTIFY messages which is simply advertising about its service.

This is also expected of Samsung TV as it would be constantly on the look out for connecting with devices for probably displaying the content they streamed. Port 3942 is also an SSDP Protocol for sending M-Search messages.

We see that the router is the device with the largest number of pre-defined ports that are involved in system functions. Because compared to any device in the network, it is the router that would be doing the largest spectrum of tasks. It's largest traffic is on port 22, which is used for SSH. This would not be the case for every router. This is rather because of the high volume of the data that we are collecting every day and the Raspberry Pi sends them to the router for sending to the remote server. We see SSH port on the source side because the remote server in-turn would give acknowledgement as SSH runs over TCP; the router sends the ACK to the Pi (which is the original sender) and this results in high traffic on port 22 on the sender's (router's) side. The next highest port number is 548 (3.8%); this corresponds to Apple File Sharing Protocol over TCP. This is a client/server protocol meant to provide file sharing and is widely used by systems running Mac OS X and use ports 548 or 427. The router must be running the AFP server to provide shared files for the many Apple clients that we have on our network. Then comes port 53 for 1.28%. This is the conventional DNS and the router is bound to have it as it must route traffic from the internal devices to the outside Internet. And for the implementation of IPSec NAT, it uses NAT-T port 4500. It also has a variety of other ports used for various functions and roles like Bootstrap server (port 67), HTTP Service (port 80), HTTPS - Secure HTTP (port 443), NetBIOS Datagram Service (port 137) etc., Thus from the presence of a very large spectrum of ports and some ports that perform the typical functions of a router like port 53 and port 67, we can clearly say that device Router (e5:a8:02) is a router.

Coming on to the non-IoT devices like Nest Thermostat, washing machine and dryer, we see a clear drop in the number of protocols used by them. Regarding all the Nest Thermostats, port 68 constitutes 100% of their port transmission data. That is

Table 6.5: Location A - Network Ports Used by IoT Devices while Transmitting

Device	Wired Medium		Wireless Medium	
	Source Port	Percentage	Source Port	Percentage
Egauge (00:27:cc)	5353	100%	5353	100%
Nest Thermostat (12:05:00)	68	100%	68	100%
Nest Thermostat (13:7a:55)	68	100%	68	100%
Nest Thermostat (14:58:68)	68	100%	68	100%
Washing Machine (34:2d:38)	68	17.39%	68	64.37%
	Custom	82.61%	Custom	35.63
Dryer (34:3d:61)	Custom	100%	68	5.41%
			Custom	94.59%

because most of the activities that the Nest is involved in like ARP, IGMP, ICMP etc., take place directly above the MAC Layer, within the LAN. There is no Transport layer operation involved in those protocols. Only bootp uses transport layer port (68). Hence we see only port 68 on Nest Thermostat devices; it represents only the Transport Layer activity. Egauge devices use port 5353% of the time. This port is used in multicast traffic as well; since Egauge is involved in sending IGMP messages, it uses port 5353 for its multicast implementation. We see that the washing machine is sending 17.39% of bootp messages (port 68), which is 2 messages over the entire 4 months. That is quite reasonable as washing machines always stay plugged in. Most of its traffic is from custom ports (82.61%). Since this is a non-IoT device, these custom ports would not be resulting from non-system applications but from Simple Service Discovery Protocol (SSDP). We see that it periodically sends SSDP NOTIFY messages, announcing its service. This is also the case with the dryer. During the period of our observation, all its traffic on the wired medium are from SSDP NOTIFY messages, advertising its services.

6.3.2 Wireless Medium

From tables 6.4 and 6.5 on pages 74 and 79 respectively, show the various ports employed by the non-IoT and IoT devices on the wireless medium. Compared to the

wired medium, we generally see a drop on the number of ports used on the wireless medium for devices like Apple_0f:fd:22, Apple_c7:eb:05, Apple_de:65:91, Router, Printer and Samsung TV; all have a reduced number of unique ports compared to the wired medium. This proves that they do more tasks on the wired medium and only fewer tasks on the wireless medium. This is quite reasonable for Apple TV, Samsung TV and Router as they are involved in data-intensive tasks like content streaming or relaying traffic between all the devices in the network (Smart TVs), facilitating their communication with the outside world etc., (Router). Hence, for them, a wired medium would be a suitable choice. Mac Mini is a stationary device and that also follows the same trend. Ipads, Laptops etc., would be used more or at least equally on the wireless medium but since these are portable, they might be taken out of the house for most of the day; their traffic is measured only when they are at home. That could be a reason why their wired ports are more in number compared to their wireless ports.

The IoT devices did not have the difference of reduced wireless ports like non-IoT devices. The tasks they do are very minimal already and there is no difference in tasks between both the mediums for them.

Thus, by observing the ports used in Transport Layer communication, one can clearly get insight into the activities that each device in the network is involved in.From our analysis, we found that non-IoT devices like Smart TVs, Ipads, Laptops,Routers etc., are involved in a greater spectrum of port numbers compared to the IoT devices. This reflects the differences in their nature. Non-IoT devices perform a wide variety of tasks and IoT devices perform fewer tasks. Then there are some particular ports that clearly point out that the device at hand is a non-IoT one. For example port 4500 implies that NAT-T is used to facilitate ESP and hence it must be a device like laptop with lot of resource at hand. Presence of port 5353 indicate that there is a good chance that the device is an Apple device which uses Bonjour

application. In addition to port 5353, use of port 7000 indicates that Apple's Quick Time Streaming Server is running on the device hence it must be a streaming device from Apple like TV. Port 8001 indicates that a Streaming Router (ST Router) is running on the device and there is a good chance that it is a content streaming device too like smart TV. Presence of a very large variety of ports compared to any device on the network by itself indicates that the device must be a router. In addition, router is the only device here with port 53, used for conventional DNS. On behalf of every device in the network, the router would be contacting the ISP for resolving IP addresses and hence this is a also good identifier for the router. Presence of port 427 indicates that Server Location Protocol is running on that device and that there is a good chance for the device to be a printer. The Nest Thermostats are the only devices here without custom ports. Because they do not give any advertisements using SSDP like washing machines and dryers. Washing machines and dryers have a good percentage of custom ports as they continuously advertise their services using SSDP NOTIFY messages. Hence a combination of the number and types of ports, along with the presence or absence of particular ports help us in identifying an IoT and non-IoT device and in some cases, even a particular device.

6.4 How many external servers do devices contact each day?

Devices present inside the network communicate with the outside world on a day to day basis. Smart TVs stream content from the Internet. Laptops, Smart phones, Ipads etc., browse the Internet by visiting various websites. They are also used to access and use cloud applications. IoT devices contact the servers present outside the network too. Their primary task is to collect data about their environment and send the collected data to an external server for processing and storage.

But the difference between IoT and non-IoT devices is that while IoT devices would always be contacting servers dedicated for receiving data from them, non-IoT

devices would be contacting numerous servers that are totally unrelated to each other. The servers contacted by IoT devices would often belong to the manufacturer of the IoT device (or in some cases, third-party data centers). But the servers contacted by non-IoT devices can be anything on the Internet and it totally depends on the user. For example, a Nest device would always send data to servers that belong to NestLabs whereas a laptop would contact the YouTube server when the user watches videos on YouTube or would contact Netflix when he is streaming a mobie from Netflix.

Hence, observing what IP external IP addresses are contacted by each device would help us in clearly differentiating between the IoT and non-IoT devices. The number of unique IPs accessed by the non-IoT devices would be much greater than that of the IoT devices. Also, from analyzing which servers is contacted, we can possibly identify the type of device.

6.4.1 Wired Medium

Tables 6.6 and 6.7 on pages 83 and 84 show the various external servers contacted by some of the devices present inside the network.

By looking at the tables we can clearly see the heterogeneity of the servers contacted by devices like Laptop, Roku, device Apple (0f:fd:22) when compared to the homogeneity of the servers contacted by Nest Thermostats, Washing Machine and Dryer. This is a clear indication of the differences between the two types of devices. IoT devices, being configured to send data to a dedicated server and don't do anything else much, would have contacted IP addresses belonging to their manufacturer predominantly (or some third party approved by the manufacturer). Non-IoT devices like laptops, smart phones, TV etc., would have contacted a variety of servers depending on the various tasks they are doing. This is verified in the results obtained.

Roku communicates with various servers that belong to Netflix Streaming Services (resulted from watching movies/series in Netflix), Cloudflare Inc. (which provides

Table 6.6: Location A - External Servers Contacted - Wired Medium - Set 1

Device	Wired Medium	
	Server IP	**Server Owner**
Nest Thermostat (2b:2b:61)	184.72.103.48	Amazon.com, Inc.
	204.236.200.82	Amazon.com, Inc.
	34.201.170.187	Amazon Technologies Inc.
	34.228.244.61	Amazon Technologies Inc.
	52.205.137.10	Amazon Technologies Inc.
	52.23.198.230	Amazon Technologies Inc.
	52.90.9.11	Amazon Technologies Inc.
	52.205.137.10	Amazon Data Services NoVa
	54.152.107.0	Amazon Technologies Inc.
	54.152.113.8	Amazon Technologies Inc.
	54.175.24.43	Amazon Technologies Inc.
	54.198.192.114	Amazon.com, Inc.
	54.205.160.220	Amazon Technologies Inc.
	54.205.194.200	Amazon Technologies Inc.
	54.227.150.20	Amazon Technologies Inc.
	54.88.25.117	Amazon.com, Inc.
Raspberry Pi 1 (b7:e0:1f)	128.119.86.173	University of Massachusetts (UNIVER-6)
	128.119.82.234	University of Massachusetts
Roku (72:1a:5d)	104.16.55.21	Cloudflare, Inc.
	199.127.194.93	Conviva, Inc.
	23.213.55.183	Akamai Technologies, Inc.
	23.246.6.*	Netflix Streaming Services Inc.
	23.246.7.*	Netflix Streaming Services Inc.
	34.231.219.195	Amazon Technologies Inc.
	45.57.45.*	Netflix Streaming Services Inc.
	52.73.138.113	Amazon Technologies Inc.
	54.154.195.110	Amazon Technologies Inc.
	8.253.154.102	Level 3 Parent, LLC
Samsung TV (e1:c2:0a)	23.215.105.98	Akamai Technologies, Inc.
	83.68.80.47	Multimedia S.A., Poland
	48.46.49.10	The Prudential Insurance Company of America, US
Washing Machine (34:2d:38)	112.106.186.252	SamsungSDS Inc.
Dryer (34:3d:61)	112.106.186.25	SamsungSDS Inc.
Egauge (00:00:b8)	128.119.245.5	University of Massachusetts (UNIVER-6)

Table 6.7: Location A - External Servers Contacted - Wired Medium - Set 2

Device	Wired Medium	
	Server IP	**Server Owner**
Apple (0f:fd:22)	151.101.186.110	Fastly (SKYCA-3)
	17.249.108.87	Apple Inc.
	17.249.124.31	Apple Inc.
	205.234.175.175	CacheNetworks, Inc.
	23.35.132.187	Akamai Technologies, Inc.
Laptop (1c:ce:28)	104.113.69.137	Akamai Technologies, Inc.
	104.20.30.2	Cloudflare, Inc.
	104.244.43.48	Twitter, Inc.
	104.88.87.78	Akamai Technologies, Inc.
	107.152.26.219	Box.com
	108.174.11.81	LinkedIn Corporation
	151.101.1.167	Fastly (SKYCA-3)
	17.249.108.27	Apple Inc.
	185.199.110.153	GitHub, Inc.
	198.252.206.25	Stack Exchange, Inc.
	199.16.156.105	Twitter Inc.
	23.192.51.85	Akamai Technologies, Inc.
	34.193.66.115	Amazon Technologies Inc.
	52.86.105.127	Amazon Technologies Inc.
	72.247.63.24	Akamai Technologies, Inc.
Ipad (41:a9:38)	17.154.67.26	Apple Inc.
	17.249.108.100	Apple Inc.
	23.197.35.150	Akamai Technologies, Inc.

Content Delivery and security services), Conviva, Inc. (online AI platform for video analytics and optimization), Amazon Technologies, Inc. (might be a result of watching Prime Videos) and Level 3 Communications (probably provides Internet Access). Not only are these servers varied but also we see that all are aligned towards some form of video access. Hence a device with mostly external servers associated with content delivery has to be a content streaming device like Roku. The same is the case for Laptop (1c:ce:28). We find that the servers accessed by this device belong to

cloud applications or web applications that are typically accessed from a laptop like Box.com, GitHub, Inc. , Stack Exchange, Inc. (commonly accessed during development phases). In addition we find that it accesses a variety of servers belonging to Fastly (content delivery and security servcies platform), Twitter, Inc. Akamai Technologies, Inc. and Amazon Technologies, Inc. Device Apple (0f:fd:22) also accesses different servers like Fastly, Apple, Inc., CacheNetworks, Inc. All these are indicators of the variety of applications run by these non-IoT devices or a variety of domains related to the main purpose of the product (like Roku). Router is the device with the highest number of external servers contacted. That is the expected result because the router is the device that communicates with every server outside the local network on behalf of the devices inside the network.

IoT devices like Nest Thermostats, Washing Machine, Dryer etc., either contact a single external server or a group of IP addresses that belong to the same provider. Washing Machine and Dryer, both contact SamsungSDS, Inc., meaning that the appliance manufacturer is Samsung. That is the only external IP they communicate with. Nest Thermostats, on the other hand, communicate with a host of IP addresses but all belong to the same organization, Amazon. This goes to say that Amazon is the third party approved by NestLabs to store their data (NestLabs could even be using the tools provided by AWS for their analytics). In a big data center like AWS, there could be a set of servers that are assigned to NestLabs and the Nest devices could have been configured to communicate any of them, whichever can serve them at the moment. Egauge devices all contact only one server, which belongs to the University of Massachusetts. It could be they are one of the many devices installed there for experimentation purposes and they were configured to communicate the University of Massachusetts server.

6.4.2 Wireless Medium

The number of external servers contacted by the devices on the wireless medium significantly decrease. IoT devices like Nest Thermostats, Washing Machines and Dryers do not contact any server over the wireless medium. All their communication over the wireless medium are related to protocols like ARP, ICMP, IGMP etc., that operate only on the IP layer and below, within the LAN.

Hence, observing the external servers contacted by each device and analyzing their degree of homogeneity would give us a clear result as to whether the device is an IoT or a non-IoT device. IoT devices would have contacted homogenous IP addresses that belong almost exclusively to their manufacturer or to a third party that was probably approved by the manufacturer. Non-IoT devices always have a variety of IP addresses in contact that belong to different organizations. Analyzing the businesses of those organizations would give us insight into the kind of non-IoT device it could be. Example, Roku's external servers all belonged to organizations that provide some form of content delivery or assist in video access. The servers contacted by the laptop were ones that are typically accesses by an engineering/software professional. Even if we are not able to identify what type of device it could be, differentiating them from IoT devices was quite easy.

6.5 Destination Devices Contacted within the network

Devices present in a LAN (Local Area Network) not only communicate with the outside but with devices present inside the network too. Observing which device(s) each device communicates with inside the network will give an insight into how devices within the LAN interact with one another.

6.5.1 Wired Medium

Table 6.8 on page 88 shows the list of destination devices that are contacted by each device on the wired interface during 1 entire data and their corresponding protocols and percentages. This does not include the multicast MAC addresses that are used by the various the protocols. The destination addresses under each source device gives an idea on what devices talk with each other directly, apart from communicating via multicasting messaging groups.

In general, Devices like Apple_0f:fd:22, Laptop (1c:ce:28), Ipad (41:a9:38), Apple TV etc., communicate regularly with Aironet_ff:ff:00 and LucentTe_00:01:00 which must be Access Points by Cisco and Alcatel Lucent respectively. Hence these devices, excluding the devices that do not contact Aironet_ff:ff:00 and LucentTe_00:01:00, must be within the Cisco and Alcatel APs and maintained by them. The interaction between Samsung TV and Laptop was purely tcp communication. That means that media from the laptop was sent to the Samsung TV for displying. On an average, 65% of the communication between Roku and Ipad (41:a9:38) were used for sending data, meaning that contents from Ipad were sent to Roku which is called as mirroring. Similarly Ipad (41:a9:3e) contacted Samsung TV (50:6a:f1) regularly for content displaying. Raspberry Pi regularly communicated with the router. Among the protocols used, more than 80% is ssh. This is reasonable because the Pi would be transferring all the collected data to the router for sending to the remote server. The files are sent using Secure File Transfer Protocol (SFTP) that operates over SSH. Devices Apple_de:65:91 and Apple_86:34:26 regularly communicate with Samsung TV to send content to the Samsung TV for displaying. The devie with the largest number of destination servers is the router. It contacted Laptop (1c:ce:28) over tcp predominantly, Ipad (41:a9:3e) over tcp, Apple TV (50:6a:f1) using data, ip among other protocols, MacMini (aa:8d:8f) over ip, data, fb_zero, Egauges over tcp, ip, etc., and NestLabs and Washing Machines and dryers using ip, tcp etc.,

Table 6.8: Location A - Destination Devices contacted within the Network on Wired Interface - 1 days' data

Source Device	Destination Devices	Protocol	Percentage(in a day)
Apple_de:65:91	Aironet_ff:ff:00	wlccp	100%
	LucentTe_00:01:00	data	100%
Router (e5:a8:02)	Apple TV (50:6a:f1)	arp	69.2%
		tcp	23%
		wol	7.6%
	Apple_86:34:26	tcp	100%
	NestLabs_12:05:00	tcp	100%
	NestLabs_13:7a:55	tcp	100%
	Apple TV (50:6a:f1)	arp	69.2%
		tcp	23%
		wol	7.6%
	Nest Thermostat (2b:2b:61)	tcp	97.7%
		data	2.2%
Apple_de:65:91	Raspberry Pi 1 (b7:e0:1f)	ssh	95.2%
		tcp	4.5%
		arp	0.1%
	Aironet_ff:ff:00	wlccp	100%
	LucentTe_00:01:00	data	100%

Table 6.9: Location A - Destination Devices contacted within the Network on Wireless Interface - 1 days' data

Source Device	Destination Devices	Protocol	Percentage(in a day)
Router (e5:a8:02)	Raspberry Pi 2 (28:4c:14)	tcp	60.9%
		ssh	37%
		arp	1.9%
		bootp	0.2%
AP (ea:b3:5e)	Raspberry Pi 2 (28:4c:14)	eapol	100%
Raspberry Pi 2 (28:4c:14)	Router (e5:a8:02)	ssh	53.1%
		tcp	45.3%
		arp	1.4%
		bootp	0.2%
Raspberry Pi 2 (28:4c:14)	AP (ea:b3:5e)	eapol	100%

6.5.2 Wireless Medium

Table 6.9 on page 88 shows the source and destination devices that communicated over the wireless interface on the same day as the data shown for the wired data. We see that the Raspberry Pi and the AP communicate regularly over ssh and tcp. This

is the interface of the Raspberry Pi 1 device that is used to remotely login to the Pi. Hence, it shows a high percentage of SSH on both the sender and the receiver side of the device. We see protocol 'eapol' being used which stands for Extensible Authentication Protocol over LAN. This was probably used to provide authentication to the remote login.

Thus by examining the destination device of each communication, we are able to get an idea about which devices within the LAN talk to each other. We see major communications involving router. That is expected as routers would be involved in every communication over wireless medium and it would be constantly communicating with other devices for network maintenance and other things. The other devices that we see talking directly to each other are the Samsung TV/Apple TV/Roku with laptops, Ipads and Iphones. These would be scenarios where content was exchanged between the devices. Also some of the laptops and Ipads regularly communicated with Cisco and Alcatel Lucent APs. Probably they are under those APs. That is why we see them communicating with the APs but not other devices. Raspberry Pi devices also regularly sent data to the router for sending to external server. IoT devices predominantly communicated via multicast messages within the LAN and did not talk to any device directly. This could be one of the markers for identifying IoT devices.

6.6 Traffic Associated with a device

The traffic associated with a device is the traffic that is both sent and received by the device. Each device contributes to the traffic associated with the network they are in by either sending or receiving packets, the basic unit of network traffic. While the header length is mostly constant across the layers of the network (unless optional fields are included), the actual data sent or received varies according to the content (payload). Earlier we examined the traffic that is associated with each device and

tried to classify them based on various parameters with the data that was available from outside the network. It yielded good results and helped us differentiate the devices as IoT and non-IoT devices and narrow down the type of device in some cases. But now we are observing the traffic from inside the network. So we should be able to get much more insights about the devices.

Also, along with examining the traffic characteristics of the devices, we would also try to group devices of similar behaviour under each parameter, so that it will help us in profiling each device.

6.6.1 How much Traffic do Devices Send and Receive ?

As we discussed in previous chapter, the traffic volume of a device (traffic that is both sent and received) would indicate whether the device is a conventional device connected to the Internet like a smart phone or a laptop or an IoT device whose traffic would be considerably less than that of the non-IoT devices. Because devices like laptops and smart phones are used for various purposes like browsing the Internet, streaming content, accessing cloud application etc., they would have a higher traffic volume associated with them. But IoT devices often would perform one dedicated task Hence their traffic would be much lower than a non-IoT device.

Just as we measured the various parameters associated with device traffic from the data we captued externally, we would do the same measure internally. We would take the same Table 5.14 on page 45 for grouping our devices based traffic volume.

6.6.1.1 Wired Medium

Table 6.10 on page 91 shows the average traffic of each device taken over 4 months. At the outset, we see more number of devices than we saw from outside the network. Because, when we are outside, packets from devices that are too far away will not reach us. But inside the network, the multiple APs present extend the network and the signal to ensure all devices wanting to communicate can easily do so. Hence we

Table 6.10: Location A - Average Traffic generated in a minute on Wired Medium

Device	Avg. of Mean(Bytes)	Traffic Category
Egauge (00:00:b8)	64.11	Very Low
Egauge (00:27:cc)	69.15	Very Low
AP (ea:b3:5e)	76.04	Very Low
Nest Thermostat (14:58:68)	97.03	Very Low
Nest Thermostat (15:3c:6e)	100.63	Very Low (Borderline)
Nest Thermostat (12:05:00)	101.7	Low (Borderline)
Raspberry Pi 1 (35:bb:d5)	102.82	Low
Insteon Hub (46:fa:a7)	142.84	Low
Washing Machine (34:2d:38)	143.67	Low
Apple Macmini (ba:b3:d6)	160.7	Low
Tp-LinkT_49:07:24	161.73	Low
Nintendo_09:28:c7	197.03	Low
Apple Ipad (0d:3e:23)	217.45	Low
Printer (08:bf:14)	279.4	Low
Roku AP (72:1a:5d)	444.83	Low
Apple Ipad (41:a9:3e)	773.34	Mid-High
Nintendo_fa:9d:30	895.43	Mid-High
Laptop (47:d9:d9)	1002.0	High (Borderline)
Apple Ipad (41:a9:38)	1179.52	High
Raspberry Pi 2 (28:4c:14)	1188.39	High
ObihaiTe_62:e9:8e	1240.0	High
Apple Iphone (85:f7:1d)	1386.83	High
Laptop (1c:ce:28)	1954.62	High
Mac Mini (aa:8d:ef)	3818.52	High
Dryer (34:3d:61)	7513.62	Very High
Samsung TV (e1:c2:0a)	11007.3	Extremely High
Apple TV (50:6a:f1)	15626.4	Extremely High
Raspberry Pi 1 (b7:e0:1f)	410221.6	Extremely High
Router (e5:a8:02)	412712.0	Extremely High

are able to see more number of devices from the inside. By looking at the table, we see a clear difference between the traffic volume of non-IoT devices like Apple TV, Router, Samsung TV, Raspberry Pi (because it collects the data and sends it over to the remote server) and IoT devices like Egauge devices, Nest Thermostats, Washing Machine etc., This is a clear marker for the two kinds of devices. Earlier, we saw that Egauge devices are only involved in IGMP multicast communication; hence, their traffic is quite low. Nest thermostats also performed relatively low number of tasks like arp, icmp, igmp and mdns. Hence, their traffic is also low. On the other hand, Smart

TVs are involved in content streaming which is very data-intensive. The router too would be relaying all the traffic between the devices to facilitate their communication. Hence these devices have extremely high traffic. It is surprising to see the dryer to have a very high traffic. But these are not data but just SSDP NOTIFY messages advertising its services. Of the two Nintendo devices, one Nintendo (09:28:c7) has very less traffic than Nintendo (fa:9d:30). This means is used more often than the other. MacMini (aa:8d:ef) has a high traffic. Earlier we saw that it was running a lot of custom ports. Hence it must be running a lot of non-system applications. Both the laptops Laptop (1c:ce:28) and laptop (47:d9:d9) also have high amounts of traffic. This is expected as laptops would be used for a variety of purposes like browsing internet, using cloud applications, etc., TP-Link has a low traffic; it means it is most probably used as an AP to extend the network and does not so any routing functions. We see that Ipad (0d:3e:23) has a low traffic. Most probably because we are measuring traffic on the wired medium and it is not connected to the wired medium often. Of the two Raspberry Pi devices, Pi 1 collects traffic in monitor mode and Pi 2 in promiscuous mode. Therefore, Pi 1 would have more data to send to the remote server than the Pi 2. Hence its traffic is higher than that of Pi 2. And the various other devices also vary in their traffic volume and fall into different categories according to their activities. Hence aveage traffic is a realiable indicator of the type of device concerned.

6.6.1.2 Wireless Medium

Table 6.11 on page 93 shows the traffic volume average of various devices calculated over the wireless medium. Egauge devices, as observed before, have a low traffic on wireless medium as well. The Raspberry Pi 2, which is used for capturing data from inside the network has 2 interfaces on the wireless medium. The interface 43:46:83 is put in promiscuous mode for passively capturing the packets. Though it

Table 6.11: Location A - Average Traffic generated in a minute on Wireless Medium

Device	Avg. of Mean(Bytes)	Traffic Category
Egauge (00:04:a4)	61.68	Very Low
Egauge (00:27:cc)	65.91	Very Low
Raspberry Pi 2 (43:46:83)	46.48	Very Low
Tp-LinkT_49:07:24	80.99	Very Low
Raspberry Pi 1 (b7:e0:1f)	62.3	Very Low
Apple Ipad (0d:3e:23)	170.58	Low
Apple Ipad (41:a9:3e)	210.08	Low
AP (ea:b3:5e)	289.96	Low
Roku_6b:2a:89	128.17	Low
Roku_72:1a:5d	389.49	Low
Insteon Hub (46:fa:a7)	142.78	Low
Washing Machine (34:2d:38)	117.77	Low
Apple Ipad (41:a9:38)	709.69	Mid-High
Nest Thermostat (12:05:00)	780.93	Mid-High
Nest Thermostat (14:48:4b)	684.67	Mid-High
Nintendo (09:28:c7)	519.43	Mid-High (Borderline)
Nintendo (fa:9d:30)	766.68	Mid-High
ObihaiTe_62:e9:8e	930.0	Mid-High
Printer (08:bf:14)	2309.61	High
Laptop (1c:ce:28)	1083.8	High
Apple Macmini (aa:8d:ef)	4799.42	High (Borderline)
Samsung TV (e1:c2:0a)	2525.95	High
Dryer (34:3d:61)	3423.14	High
Router (e5:a8:02)	12326.51	Extremely High
Raspberry Pi 2 (28:4c:14)	11519.33	Extremely High
Apple TV (50:6a:f1)	15320.79	Extremely High

captures packets from all the devices, the packets intended for it (packets sent with this particular interface as the destination address) would be very less and it hardly sends out any packet as well. Hence its traffic is very low. But interface 28:4c:14 is used for other communication and for logging in from the remote server; therefore that interface has higher traffic. Tp-link is a wireless router. From the low traffic on the wireless interface, it is probably used as a wireless distribution system, a system that interconnects APs in a network eliminating the need for a wired backbone. Raspberry Pi 1 is the device that is used to capture the traffic externally in monitor mode. It sends the collected data via the wired medium. Because we could see a high traffic for this device on the wired medium (It is common for interfaces on different medium on

the same device to have the same MAC address; since the interfaces will never overlap, this is not a problem). In that case, it would not be used much for anything else and hence, the low traffic. Ipads 0d:3e:23 and 41:a9:3e and devices Apple_61:b1:70 and Apple_c7:eb:05 all have low traffic. This could either be because these devices are not used very much or they could be a little far away from the point of data capture that not all packets from them are captured successfully. Roku devices also show a low traffic; may be devices like Apple TV and Samsung TV are used more than the Roku and hence the low traffic. Because even on the wired interface, Roku device has a very low traffic. Insteon Hubs are devices used to interface with the many smart devices present at home and access them remotely via smart phones. Based on its purpose, its traffic would be in sporadic short bursts when it communicates with another smart device for any action to-dos; there is no other content involved and hence the low traffic. Nest Thermostats, being IoT devices, would naturally have a traffic lesser than conventional devices since they perform only limited set of tasks. We can see that reflected on the wireless medium as well. Mac Mini (aa:8d:ef), as expected has a high traffic. Printer has a higher traffic on the wireless medium than the wired. This is reasonable because documents for printers would often be sent from devices like smart phones and laptops that connect to the printer via the wireless medium. Hence the high wireless traffic. Thus the variations in the mean value of Traffic volume gives us a clear idea on what kind of device it could be.

Comparing this result with the result that we got from analysis of the external data, though each device does not fall under the same category of Traffic Volume as it did on the external data, we can see that the relative ordering of the devices is largely unchanged. We are measuring both the categories with the same scale, but internally we would capture more data as we are inside the network. Hence, the magnitude of the data might be larger internally and the categories might be different, but still

each is effective in differentiating IoT and non-IoT devices in the point where the data is captured - internal or external.

6.6.2 Do Devices consistently send the same amount of traffic everyday?

In the previous section, we saw how much traffic devices send and receive on an average. But that does not give us any idea about the fluctuations in traffic. For example, as we saw in the previous chapter, the usage of a non-IoT device would vary day to day, from user to user. As these devices have human intervention and assist the user in their daily tasks, their traffic is swayed by the user. For example, a smart TV's traffic would largely depend on how much the occupants of the house watch TV. And it is highly unlikely that they watch exactly the same amount of TV everyday. But IoT devices, on the other hand, require little to no human intervention; therefore their traffic largely depends on their environment. Hence unless there is a drastic change in the environment, their behaviour will not vary. Eg: Nest Thermostat radically changes its degree of activity when the winter passes and spring/summer comes; but until then it is mostly consistent in its behaviour. Therefore measuring how much a device's traffic sways could be a good indication of its nature.

We would use the same Table 5.18 on page 52 that we used for the external data, for grouping our internal data too.

6.6.2.1 Wired Interface

Table 6.12 on page 96 shows the variance values of the average traffic for various devices and their fluctuation categories.

We can see that devices with low traffic fluctuation i.e., small standard deviation from the mean traffic are the IoT devices like Egauges, Nest Thermostats and Washing machines. Egauges fall under Level 1 Fluctuation Category whereas Nest Thermostats and washing machines fall under Level 2. This means that their traffic consistently stayed within a very low swaying margin over the days that we captured the data.

Table 6.12: Location A - Fluctuations in Traffic on Wired Medium

Device	Avg. of Mean	Standard Deviation	Variance	Fluctuation Category
Laptop (47:d9:d9)	1002.0	0.0	0.0	Level 0
ObihaiTe_62:e9:8e	1240.0	0.0	0.0	Level 0
AP (ea:b3:5e)	76.04	3.56	100.77	Level 1
Egauge (00:00:b8)	64.11	15.02	310.46	Level 1
Egauge (00:27:cc)	69.15	28.34	868.98	Level 1
Raspberry Pi 1 (35:bb:d5)	102.82	83.65	34739.28	Level 1
Raspberry Pi 2 (43:46:83)	64.87	36.35	10322.12	Level 1
Insteon Hub (46:fa:a7)	142.84	69.61	6206.27	Level 1
Apple Ipad (0d:3e:23)	217.45	183.69	55422.89	Level 2
Apple MacMini (ba:b3:d6)	160.7	181.48	35852.1	Level 2
Nest Thermostat (12:05:00)	101.7	139.4	134261.11	Level 2
Nest Thermostat (13:7a:55)	93.92	114.34	118014.85	Level 2
Nest Thermostat (14:48:4b)	111.1	180.49	223682.08	Level 2
Nintendo_09:28:c7	197.03	171.28	49826.3	Level 2
Nintendo_fa:9d:30	895.43	114.57	44026.14	Level 2
Washing Machine (34:2d:38)	143.67	129.6	46684.66	Level 2
Tp-LinkT_49:07:24	161.73	273.5	2963968.0	Level 3
Roku_72:1a:5d	444.83	699.09	618832.0	Level 7 (Borderline)
Ipad (41:a9:3e)	773.34	862.14	1795364.41	Level 9
Iphone (85:f7:1d)	1386.83	984.83	1027020.12	Level 10
Laptop (1c:ce:28)	1954.62	1661.88	4699611.11	Level 11
Ipad (41:a9:38)	1179.52	1167.95	2060067.23	Level 11
Printer (08:bf:14)	279.4	1065.29	1548739.84	Level 11
Raspberry Pi 2 (28:4c:14)	1188.39	2164.68	17180696.97	Level 12
Samsung TV (e1:c2:0a)	11007.3	4526.25	21387478.26	Level 14
MacMini (aa:8d:ef)	3818.52	4800.43	34065583.33	Level 14
Apple TV (50:6a:f1)	15626.4	7397.75	56813920.0	Level 17
Router (e5:a8:02)	412712.0	1595776.0	7640400000000.0	Level 20
Dryer (34:3d:61)	7513.62	18833.97	2113305785.12	Level 20
Raspberry Pi 1 (b7:e0:1f)	410221.6	1595864.0	7641592000000.0	Level 20

Because in the previous sections, we saw that they were only involved in regular multicast messaging and ARP messages and did not involve in any data related task. For Nest Thermostats it could be because that the data was collected during the summer months and they did not do much. Hence there is nothing to sway their traffic. Washing machine sends SSDP traffic from custom ports but we see that it is regular in its advertisement unlike the dryer. We know that the dryer sends SSDP NOTIFY messages from the previous sections. From the high fluctuation, we can conclude that it does not advertise in the same rate everyday. Nintendo devices also have a high fluctuation. That is expected because its traffic would vary depending

on how much it is used and it is unlikely that it is used for the same amount of time everyday. Printer (08:bf:14) also has a high fluctuation. This is expected because it would receive documents for printing only sporadically. The rest of the devices like Samsung TV, Apple TV, Router all have highly fluctuating traffic. Because the Smart TVs would vary depending on usage and the router's traffic would always depend on the traffic of its internal devices. The Raspberry Pi 1 (b7:e0:1f) would also have varied traffic depending on the devices inside the network as it is only sending the data it captured from the devices inside. Laptops, Iphones and Ipads all have high traffic fluctuation as they would also have varied traffic depending on usage. The same is the case for MacMini too. Thus the average traffic, combined with how much the traffic fluctuates could easily help us in differentiating IoT and non-IoT devices.

6.6.2.2 Wireless Interface

Table 6.13 on page 98 shows the traffic fluctuation for various devices under wireless medium. We see the same scenario reflected on the wireless medium also. Devices like Egauge, Nest Thermostats, Washing machine all have low traffic fluctuation as expected. And devices like router (e5:a8:02) and Apple TV (50:6a:f1) have highly fluctuating traffic. We see that Samsung TV (e1:c2:0a) has a low fluctuation on the wireless medium. This must be resulting from the device using the wired interface more than the wireless interface; contents to be streamed are fetched via the wired interface. This is proven in the previous section, where we can see that the average traffic of Samsung TV on wired is much higher than the average traffic in wireless medium. Raspberry Pi 1 (b7:e0:1f) is the built-in interface of the Pi and it is not used much, consistently. Hence its traffic as well as fluctuation is low. Raspberry Pi 2 (28:4c:14), the interface that is used primarily for logging in from the server, has a high fluctuation. This is reasonable as I do not login everyday. Sometimes I login once a week and close the connection immediately. Sometimes I keep the connected

Table 6.13: Location A - Fluctuations in Traffic on Wireless Medium

Device	Avg. of Mean	Standard Deviation	Variance	Fluctuation Category
Nest Thermostat (15:3c:6e)	632.0	0.0	0.0	Level 0
Nest Thermostat (2b:2b:61)	632.0	0.0	0.0	Level 0
Ipad (0d:3e:23)	170.58	90.74	14915.33	Level 1
AP (ea:b3:5e)	289.96	57.69	5866.26	Level 1
Egauge (00:00:b8)	61.51	8.32	95.22	Level 1
Egauge (00:27:cc)	65.91	22.24	540.28	Level 1
Raspberry Pi 2 (43:46:83)	46.48	3.55	22.0	Level 1
Nest Thermostat (12:05:00)	780.93	75.52	28515.4	Level 1
Nest Thermostat (14:58:68)	595.08	53.48	8704.83	Level 1
Nintendo (fa:9d:30)	766.68	88.91	28932.97	Level 1
Raspberry Pi 1 (b7:e0:1f)	62.3	15.57	3111.44	Level 1
Roku_72:1a:5d	389.49	57.24	11255.04	Level 1
Insteon Hub (46:fa:a7)	142.78	77.08	9422.6	Level 1
Tp-LinkT_49:07:24	80.99	44.99	2443.32	Level 1
Nest Thermostat (14:48:4b)	684.67	100.45	40362.25	Level 2 (Borderline)
Samsung TV (e1:c2:0a)	2525.95	190.87	41983.9	Level 2
Washing Machine (34:2d:38)	117.77	101.04	32454.39	Level 2 (Borderline)
Nintendo_09:28:c7	519.43	300.26	90154.3	Level 3 (Borderline)
Roku_6b:2a:89	128.17	224.35	60428.57	Level 3
Ipad (41:a9:3e)	210.08	352.6	840862.79	Level 4
ObihaiTe_62:e9:8e	930.0	395.98	313600.0	Level 4
Ipad (41:a9:38)	709.69	690.36	603082.5	Level 7
Laptop (1c:ce:28)	1083.8	829.25	1209028.95	Level 9
Printer (08:bf:14)	2309.61	2211.96	7435017.54	Level 12
Macmini (aa:8d:ef)	4799.42	5186.6	32349473.68	Level 15
Apple TV (50:6a:f1)	15320.79	7423.67	58018771.93	Level 17
Dryer (34:3d:61)	3423.14	7055.75	54827368.42	Level 17
Router (e5:a8:02)	12326.51	100515.96	22020350877.19	Level 20
Raspberry Pi 2 (28:4c:14)	11519.33	102765.96	22789473684.21	Level 20

session open for hours and days together. Hence the high fluctuation. The dryer also has highly fluctuating traffic. Printer (08:bf:14) also has a highly varying wireless traffic. As said before, this totally depends and when and how much documents are sent to it for printing over the wireless interface. Therefore the fluctuation rates of each device, both on its wired and wireless interface, will give us a unique foot print of each device based on its behaviour in both the interfaces combined. Nintendo (fa:9d:30) has Level 1 variance on the wireless interface but had Level 2 variance on the wired interface. That means it's traffic fluctuated more on the wired interface comparatively. But there does not seem to be a big difference in the usage between its

wired and wireless interfaces as both have almost the same average traffic and traffic does not fluctuate largely. But Nintendo (09:28:c7) had a low traffic on wired medium and mid-high traffic on wireless and a fluctuation category of Level 2 on wired and Level 3 on wireless; this means it did not use its wired interface consistently and wireless has been used now and then with large variations. This difference in the traffic average and its variance can be used to distinguish between the two Nintendo devices. Similarly, every device would have a unique combination of traffic mean and its variance according to usage. This could be taken advantage of when trying to identify individual devices.

6.7 Do Devices only send data or receive data or both ?

In the previous chapter we tried to analyze the traffic direction of the devices using the data we got from outside the network. We would try to evaluate the devices again under the same criterion but with data collected from inside the network. Analyzing whether a device has only outgoing data or incoming data or both would give us one more factor to differentiate the IoT and non-IoT devices and then further identify the devices themselves possibly. Conventional non-IoT devices must have mixed traffic depending on their usage and the applications they run. Devices like laptops and smart phones may have varying outgoing and incoming traffic depending on their activities. On days when movies are watched, their incoming traffic would be high. On days where they interact with cloud applications and upload large files, their outgoing traffic would be high. IoT devices may not consistently send and receive traffic in equal amounts at all times. They are designed to collect and send data to servers and at the same time they also participate in Link Level protocols like arp, igmp etc., Sometimes they may send more traffic than they receive via tha LAN protocols. Hence, analyzing the direction of predominant traffic of a device would

prove to be another factor in differentiating various devices. It would also give us better understanding about the behaviour of the device.

6.7.1 Wired Interface

Table 6.14 on page 101 shows the percentage of days where each device was sending both in and out traffic or only outgoing traffic or received only incoming traffic.

First taking the Nest Thermostats, we find that they have mixed traffic on most of the days i.e., they both send and receive data on more than 80% of the days. We know that on the wired medium they have sent ARP requests, ICMP packets, IGMP packets, participated in multicast DNS, sent data to external server etc., On the receiver side, they have received acknowledgements for the packets they sent to the external servers, received messages from the multicast group. Router was exclusively having mixed traffic on all days; this is as expected because router relays the communication between all devices and hence would have mixed traffic. Raspberry Pi 1 (b7:e0:1f) also has mixed traffic because it would be sending the collected data as well as communicate with other devices on the network. For Ipads and laptop (1c:ce:28), days where they had exclusively outgoing traffic were higher than the days where they had mixed traffic. This could be because on some days, their wired interface were not used for anything but sending out media content to the smart TVs for displaying. Egauge devices also have more days where they just send data than the days where the traffic is mixed. That means, on some days, they send only the collected data to server and do not do anything much. Washing Machines and Dryers also have high percentage of exclusively outgoing traffic. This might most probably because they were simply adversitising their services on most of the days without doing anything else.

Table 6.14: Location A - Classification based on Traffic Direction - Both In and Out, Only Out and Only In - On Wired Medium

Device	Both In and Out(%)	Only Out(%)	Only In(%)	Direction Category
Router (e5:a8:02)	100.0	0.0	0.0	Exclusively Mixed
Raspberry Pi 1 (b7:e0:1f)	100.0	0.0	0.0	Exclusively Mixed
Ipad (0d:3e:23)	40.0	60.0	0.0	Only Out > Mixed
Laptop (1c:ce:28)	47.71	52.29	0.0	Only Out > Mixed
Apple TV (50:6a:f1)	57.94	42.06	0.0	Mixed > Only Out
Iphone (85:f7:1d)	55.56	44.44	0.0	Mixed > Only Out
MacMini (aa:8d:ef)	54.55	45.45	0.0	Mixed > Only Out
Raspberry Pi 2 (28:4c:14)	51.52	48.48	0.0	Mixed > Only Out
Ipad (41:a9:38)	28.33	71.67	0.0	Only Out ≫ Mixed
Ipad (41:a9:3e)	3.36	96.64	0.0	Only Out ≫ Mixed
Egauge (00:00:b8)	29.37	70.63	0.0	Only Out ≫ Mixed
Egauge (00:27:cc)	0.79	99.21	0.0	Only Out ≫ Mixed
Raspberry Pi 2 (43:46:83)	6.06	93.94	0.0	Only Out ≫ Mixed
Nintendo_fa:9d:30	4.29	95.71	0.0	Only Out ≫ Mixed
Roku_72:1a:5d	30.16	69.84	0.0	Only Out ≫ Mixed
Samsung TV (e1:c2:0a)	13.79	86.21	0.0	Only Out ≫ Mixed
Washing Machine (34:2d:38)	4.2	95.8	0.0	Only Out ≫ Mixed
Dryer (34:3d:61)	5.74	94.26	0.0	Only Out ≫ Mixed
Nest Thermostat (12:05:00)	86.51	0.0	13.49	Mixed ≫ Only In
Nest Thermostat (14:48:4b)	84.13	0.79	15.08	Mixed ≫ Only In, Negligible Only Out
Nest Thermostat (14:58:68)	83.2	0.0	16.8	Mixed ≫ Only In

6.7.2 Wireless Interface

The anlysis done on the wireless interface regarding traffic direction did not yield substantial results to differentiate the IoT and non-IoT devices. Most of the devices had exclusively outgoing traffic. That might be a result of the devices being relatively less active on the wireless medium when compared to the wired medium and not sending out any actual data. Results shown on tables 6.10 and 6.11 on pages 91 and 93 show the stark contrast in the average of the traffic for each device. In wired medium, the increase in traffic from device to device was gradual as almost all the devices were fully active over the wired medium. But increase in traffic is very abrupt on the wireless medium, with most of the devices having significantly lower traffic mean and the only devices with high traffic on the wireless medium were Apple TV, Router, Raspberry Pi 2 (28:4c:14) (which is used to communicate with the Pi

remotely) and surprisingly, dryer (Because of its high amount of SSDP messages). Hence, this section did not yield sufficient results.

6.8 Conclusion

Thus in this chapter, we examined what could be learned about the IoT devices and other devices while observing the traffic from inside the network. We were able to learn a lot about the devices using information got from protocols, network ports and external IP addresses they contacted. From the protocols and the ports, we came to know exactly what type of activity was undertaken by each device and how those activities differ between IoT and non-IoT devices. The external servers contacted by each device gave us an idea about the nature of their interaction with the devices outside the network. From the destination devices contacted by each device, we were able to learn about the interactions between the devices within the LAN. Then we measured the various network characteristics like traffic volume of each device, how their traffic fluctuates, what is their predominant traffic direction etc., Under each of those parameters, we were clearly able to see how IoT devices from non-IoT devices and how, by the presence of certain features, we could pin point to one device. Using this information, we can create a profile for each device with the information we have from how they fared under each parameter. Such profiles can help us in understanding whether the new devices that we would encounter in the future are IoT or non-IoT devices and and even help identify the specific device. That is the next step in this thesis.

CHAPTER 7

CLUSTERING

Now that we have analyzed the data that is got from both inside the outside and outside the network, we would like to group similar devices together automatically, using clustering.

Clustering is a form of unsupervised learning where in data sets are grouped together into various clusters so that similar items belong to one group. It is unsupervised because we do not say explicitly what makes a set of data points similar. This makes unsupervised learning a better choice for this case, as the algorithms will try to cluster the devices based on the innate characteristics that is present in the data sets of the devices.

We choose to use K-Means, as it is one of the simplest and fastest clustering algorithms. We have data from three sections namely External Wireless, Internal Wired and Internal Wireless. We would perform clustering on each section of the data to see how devices fare in them. We would also compare the clustering results between the External and Internal data to determine how good is the external data when compared to the internal data.

This would help us in,

- Learning the network signature strength of different devices - thereby help us in identifying different group of devices and even identify particular devices
- Investigate the difference in the network signature of devices between the Internal and External data capture - this would help us in establishing whether the

external data capture contains enough data to still obtain a good signature for devices that have a strong network signature internally also

- Enable profiling of smart homes - just by looking at the data that is got externally on the outside of the network, anyone would be able to profile a smart home, meaning that they would be able to identify the different groups of devices present in the home

7.1 Defining Network Signature

Since we would like to learn the network signatures of various devices, we would first define a network signature. A network signature shows how strong is the presence of the device in the network and it is measured from the device's network data. Hence, as we experimentally increase the number of clusters, we are looking for the following signs,

- When do similar devices emerge together as a cluster ?
- How persistent are the similar devices in staying together in a cluster ?
- What is the degree of exclusivity in the cluster - are only those similar devices present or are other devices present ? If other other devices are present, to what degree ? What is the percentage of these similar devices in that cluster ?

Hence going by that definition, devices that have a stronger signature would go into single clusters very early. And there is a higher probability that those single clusters would be exclusive for devices with stronger signatures. They would also stay in that exclusive cluster for longer. But devices that have a weak signature on the other hand, will go into single exclusive clusters very late. Also they will not stay longer in those clusters; they would break apart much easily.

7.1.1 Quantifying Signature Strength

We came up with a simple formula to quantify the network signature. This is a way to qualitatively visualize the network signature of the devices. Figure 7.1 on page 105 shows the formula for calculating signature strength of various devices.

$$\text{Signature Strength} = \frac{\text{Number of Clusters at Exclusivity}}{\text{Convergence Attained}}$$

Figure 7.1: Formula for calculating Signature Strength

In our experiment, as we increase the number of clusters, we take the number of clusters at which the device at hand (includes all devices of a particular kind, eg: all Nest Devices) becomes an exclusive cluster. To really appreciate how early or late that step is, we also take into account when the clustering reaches convergence i.e., the devices would no longer separate into new clusters.

7.1.2 Identifying Parameters Unique to a device

If we want to create a profile for each kind of device, it is important to identify the top parameters that contribute to the shift in the clusters when the devices go into exclusive clusters. These are the parameters that are unique to that device and constitute the profile of the device. To identify those parameters, we take the centroids of the clusters where these devices become exclusive and the centroid of the same cluster in the previous iteration. We find the squared difference between the two clusters and identify the parameters that have the highest difference. These are the parameters that pushed the devices to go into an exclusive cluster. For our experiment, we identify the top 5 parameters for each device.

Figure 7.2: Identifying Parameters Unique to a device

7.2 Input Data to the Clustering Algorithm

7.2.1 External Wireless

On the external data capture, data from 292 devices were observed. Since we are capturing the data in monitor mode, without associating with any network, this data includes data from all the devices that are within range of the capture, irrespective of the network they belong to. The parameters that were considered are:

- Traffic Volume
- Fluctuation in Traffic Volume
- Outbound and Inbound Traffic Percentage
- Frequency of State Change of power (sleep/awake)
- Fluctuation in Frequency of State Change of power

The input data dimension for external data is 292 x 9. The data is normalized before feeding to the algorithm.

7.2.2 Internal Wired and Wireless

On the internal data capture, data from around 69 devices were observed on the wireless interface and data from around 50 devices were observed on the wired interface. Since we are observing the data internally in promiscuous mode, this includes data from only devices that are present in the host network. The parameters considered are:

- Average Traffic Volume
- Fluctuation in Traffic Volume
- Outbound and Inbound Traffic Percentage
- Number of Type of Protocols

- Number of Type of Ports

The input data dimension is 50 x 70 for Internal Wired and 69 x 74 for Internal Wireless. For combined internal clustering is 50 x 144. The data is normalized before feeding into the algorithm.

7.3 Cluster Progression - External vs. Internal

From the clustering done on the external data, we can see that the devices are divided into various clusters based on the traffic data. Because externally, the data observed primarily consists of traffic data and power management information. But on the other hand, the internal data consists of a lot of device specific data. Therefore, when the clusters divide initially, they are based on this device specific information rather than the traffic data. Also, by using the squared difference between the centroids of successive iterations, we were able to identify precisely which parameters induced the shift towards a particular cluster. Those parameters are the characteristic features of the device.

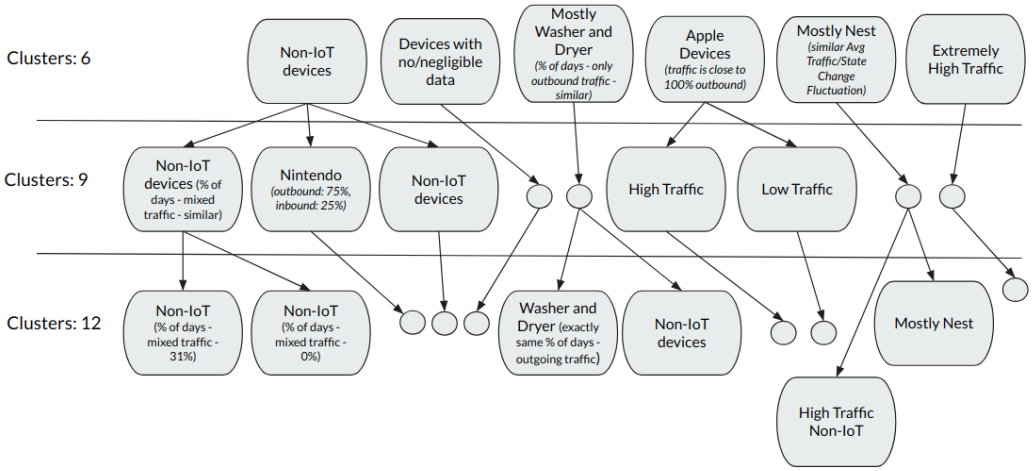

Figure 7.3: Clustering Progression of External Data

Figure 7.3 on page 107 shows the progression of clustering on the external data. At the beginning, when the number of clusters are 6, we can see that the devices are

grouped into clusters based on some attribute of their traffic. The washing machine and dryer go into separate clusters because the percentage of days where they had completely outbound traffic are similar. The Apple devices are grouped together because their outbound traffic is close to 100%. All the Nest Devices are grouped as one because their average traffic as well their fluctuation in the frequency of state change (sleep/active) are similar, extremely high traffic devices come under one cluster and one cluster has devices that have no/negligible observed data and one more cluster has non-IoT devices. When the number of clusters are increased to 9, the cluster that had non-IoT devices further breaks into non-IoT devices whose percentage of days where they had mixed traffic are similar, Nintendo devices that have similar inbound and outbound percentage and the rest of the devices in that cluster. The Apple devices break into two clusters with high and low traffic respectively. Further when he number of clusters are increased to 12, the clusters further break based on parameters that turn out to be exactly same for some devices and not so same for other devices.

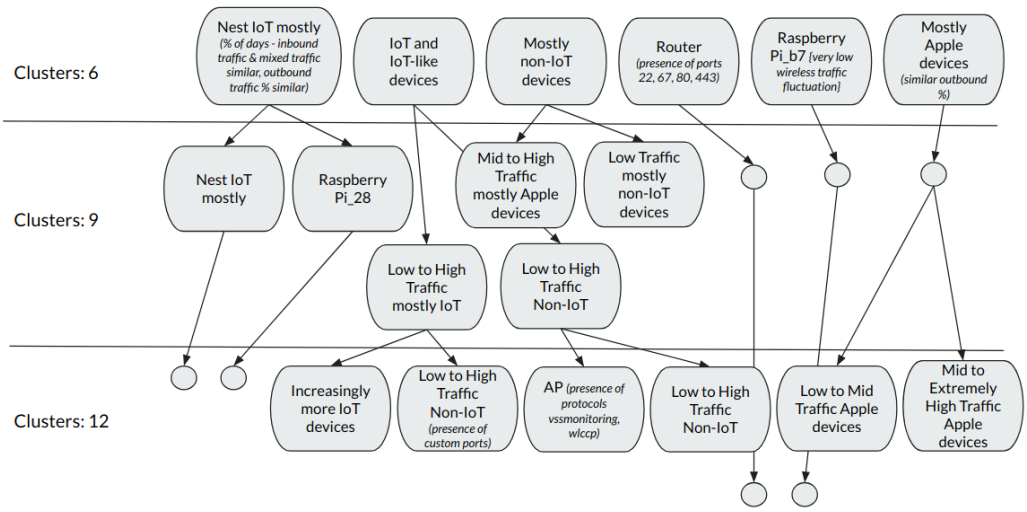

Figure 7.4: Clustering Progression of Internal Data

Figure 7.4 on page 108 shows the clustering progression of the internal data. At the outset, we can see that that initial clustering are not based on just traffic volume but

rather on device-specific parameters. When the number of clusters are 6, all the Nest devices come under a cluster with some other devices. Then there is another cluster with devices like Egauges, washing machine, dryer and some more devices. Then another cluster has exclusively router in it. The router goes into a separate cluster very early because of the presence of device-specific markers like presence of ports 22(SSH), 67(Bootp server), 80(HTTP), 443(HTTPS) etc., Raspberry Pi also goes into a separate cluster because it uniquely has a very low wireless traffic fluctuation. Many of the Apple devices come under one cluster which has predominantly Apple devices; they all have similar outbound traffic percentage. When the number of clusters are increased to 9, the cluster with predominantly Nest devices breaks into two clusters with mostly Nest and a Raspberry Pi device. Cluster with Egauges, Washers, Dryers and others break into two clusters with mostly IoT and non-IoT devices respectively. The Router and the Raspberry Pi continue as such. The cluster that had mostly Apple devices further divides into two clusters with low to mid traffic Apple devices and mid to extremely high traffic Apple devices. On successive iterations, the devices further move into more specific clusters. But the main difference between the external and internal cluster progression is that externally the initial clusters and the subsequent grouping were based on traffic parameters and power management state change data. Internally, the initial clusters were formed based on the device specific data that distinguishes types of devices readily. This data is not available externally.

7.4 Internal Wired

On the Internal Wired medium, there are around 50 devices. To give equal weight to all the parameters, we normalize the data and feed them into the algorithm. At the first iteration itself, we see that the devices are grouped into more or less meaningful clusters - non-IoT devices with a mid-range traffic, devices with low traffic and fluctuation (both IoT and non-IoT devices), non-IoT devices with extremely high

traffic etc., And as the number of clusters increase, the devices are grouped into more and more specific clusters. One note-worthy observation throughout the iterations in Internal Wired devices is that, beginning from when the number of clusters are 9 to the iteration where the number of clusters are 37, all the Nest Thermostats, and Nest Thermostats alone, are present in one cluster. Though other devices increasingly form clusters that solely consist of that device alone, right from when the number of clusters are 9, the Nest Thermostats all stay exclusively in one group until the number of clusters reach 37. This goes to show that all the Nest Thermostats are very similar in their behaviour and they have a very strong signature in the network. Figure 7.5 on page 110 illustrates the clustering progression of the Nest devices on the wired medium. It shows how Nest has a strong signature here as initially, its clusters have more than 75% Nest. Then it moves into exclusive cluster quickly.

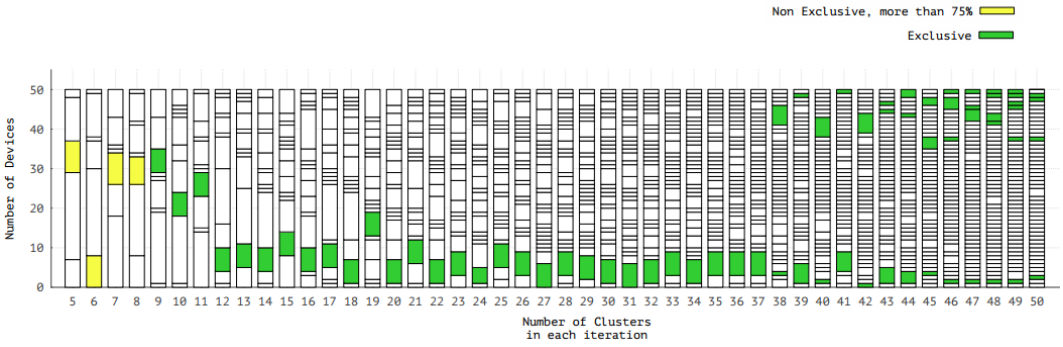

Figure 7.5: Clustering Progression of Nest Thermostats - Wired medium

Other IoT devices like Egauges, Washing Machine and Dryer were in a single cluster initially. Then, when clusters are 7, three of the Egauge devices (00:00:b8, 00:04:04, Mosberge_0a:28) were clustered together (not exclusively - had other devices in the cluster too) and one Egauge (00:27:cc), Washing machine and Dryer were in a separate cluster together (not exclusively). When clusters are 12, the initial three Egauge devices went into an exclusive cluster (no other devices). These three Egauge devices remain in this exclusive cluster until the iteration where the number of clusters

are 40. Hence, these Egauge devices also have a signature in the network. But their signature is not as strong as that of the Nest Thermostats.

The other Egauge device, Washing machine and Dryer continue to remain in a cluster with other non-IoT devices. This trend continues and when the clusters are 19, the Washing machine and Dryer go into a separate cluster (not exclusively) and the other Egauge device goes into another cluster (not exclusively). When clusters are 30, the other Egauge device goes into a cluster of its own (solely has that device alone) and the Washing Machine and Dryer remain in a separate cluster (not exclusively). When the number of clusters are 36, Washing machine goes into a cluster of its own. When the number of clusters are 44, Dryer goes into a cluster of its own.

Devices like Router and Raspberry Pi, which are very unique in the network in terms of network traffic (both very high traffic) and in terms of their activity (Router - highest spectrum of tasks in the network, Raspberry Pi - single purpose which is to collect and send data) very quickly go into clusters of their own, right from the iteration when the number of clusters are 6. Samsung TV goes into an exclusive cluster from when the number of clusters are 11. From this we can infer that these devices are very unique in the network and hence quickly form exclusive clusters.

From this clustering, we can clearly see that devices that exhibit a very strong signature like Nest Thermostats and Egauge devices (though not all) stay in exclusive clusters through most of the iterations. Other unique devices like Router, Raspberry Pi, Samsung TV also exhibit their uniqueness in the network by going into exclusive clusters very early in the iterations.

7.5 Internal Wireless

On the Internal Wireless medium, there are around 69 devices. Comparing the results of clustering on these devices to the results from the Internal Wired devices clustering, we find that the Nest Thermostats do not have a strong signature on the

Wireless medium as they had on the Wired medium. Up until when the number of clusters are 27, the Nest Thermostats stay in a cluster with other non-IoT devices. After that, they increasingly break into various clusters, either exclusively or non-exclusively. Figure 7.6 on page 112 illustrates the clustering progression of the Nest devices on the wireless medium. Initially it stays in a cluster where only 20% to 50% of the devices are Nest. Then gradually moves into successive clusters where more and more devices are Nest until they finally move into a cluster where 100% of the devices are Nest.

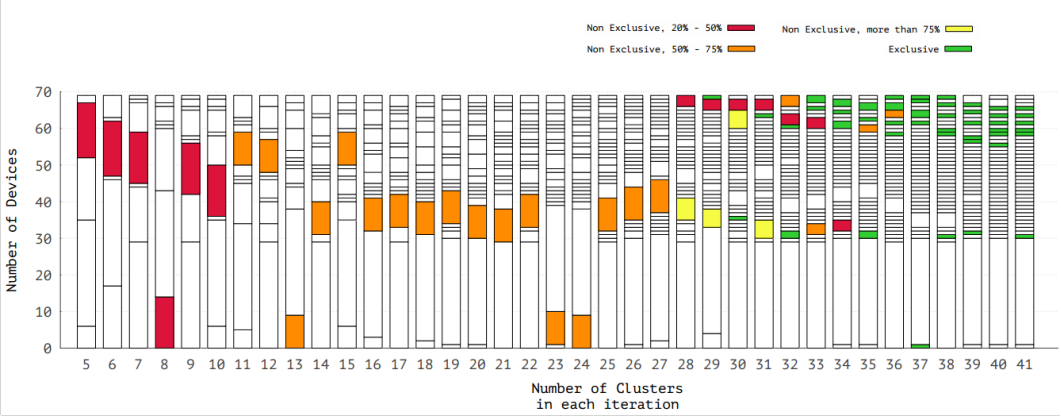

Figure 7.6: Clustering Progression of Nest Thermostats - Wireless medium

All of the Egauge devices remain in a cluster with other devices in the beginning. When the number of clusters are 14, the same three Egauge devices (00:00:b8, 00:04:04, Mosberge_0a:28) fell into a cluster (not exclusively) while the other Egauge device (00:27:cc) went into another cluster (not exclusively). This trend continues and when the number of clusters are 26, the other Egauge device goes into an exclusive cluster. The three Egauge devices continue to remain in a cluster (not exclusively) and then break apart increasingly into different clusters starting from when the number of clusters are 39. Hence both the Nest Thermostats and the Egauge devices, though they have a signature on the Wireless interface, it is not as strong as their signature on the Wired Interface. On the Wired interface, they formed exclusive clusters

(Egauges not completely) and stuck to it through a large number of iterations. The same behaviour is not exhibited on the Wireless Interface. Hence, it can be inferred that both the Nest Thermostats and the Egauge devices are more active over the Wired interface and hence, their signatures are stronger on Wired.

Washing Machine and Dryer initially remain in a cluster along with the Egauge devices. When the number of clusters are 10, they move into another cluster with some other non-IoT devices. When the number of clusters are 19, they move into a cluster of their own. When the number of clusters are 24, they move into two exclusive clusters. On the Wireless interface, they moved into an exclusive cluster sooner than they did on the Wired Interface. That is probably because, their traffic is higher on the Wired medium; the number of protocols they use over the Wired medium is 5 whereas the number of protocols they use on the Wireless medium is just 3. Hence the overall complexity of the data on the Wireless for these devices is lesser than the complexity of the data observed on the Wired Interface. Hence, they converge much quickly over the Wireless.

Samsung TV is initially present with other devices. Then it moves into a cluster with Tp-Link when number of clusters are 8 and remains there. Then it moves into an exclusive cluster when the number of clusters are 14. The router and Raspberry Pi move into an exclusive cluster much early, when the number of 6. Hence this proves again that these devices are very unique in the network.

7.6 External Wireless

For the external wireless clustering, we first take into our experiment all the devices that are observable at this point, irrespective of the network they belong to. We take a total of 292 devices, from all the networks that are within range. External clustering holds so much importance from our thesis point of view because, all these

data are available to just anyone who can passively observe the network traffic from outside, without even joining the network.

The Nest thermostats are present in a cluster with other non-IoT devices initially. When the number of clusters are 20, all the Nest thermostats move into an exclusive cluster. And they remain that way until the number of clusters are 57. When the clusters are 58, they break into two exclusive clusters consisting of 3 Nest thermostats each. They remain that way for some 5 iterations and after that they break into more exclusive clusters. By the time the clusters are 71, each of the Nest thermostats are in a unique cluster of their own, though there are still clusters that have 70+ devices in them. Hence this goes to show that, even from the data that is leaked outside the network, we can see that the Nest thermostats have a strong signature.

Egauge devices did not have a discernible signature on the external wireless data, at least when performing the clustering with all the devices that are observable at this point. Up until the last iteration where the number of clusters are 131, all the Egauge devices remained in a cluster with some 70+ devices.

Washing machine and Dryer initially remained in a cluster with some 20+ devices. Then they gradually broke into a cluster with 5+ and fewer devices. When the clusters were 37, they both moved into an exclusive cluster. And until the last iteration, the Washing machine and Dryer remain in this exclusive cluster. This goes to show that, externally, the Washing machine and Dryer have stronger signatures when compared to the Egauge devices.

The router, from the beginning till the end, remains in a cluster with some 70+ devices. Hence, the router does not have a discernible signature at the external wireless data, at least when performing clustering with all the external devices. But the Access Point that we have observed throughout our external data analysis, Apple (ea:b3:5e) has a very strong signature. It goes into an exclusive cluster much early, when the clusters are 6. And it remains in this exclusive cluster till the last iteration.

Hence, we can conclude that while we managed to capture most of the data from the AP at this external point, we did not capture as much data from the router.

Samsung TV initially remains in a cluster with other devices. When the clusters are 20, it moves into a cluster with Roku and when the clusters are 36, it moves into an exclusive cluster of its own. It remains in this exclusive cluster till the last iteration. Hence Samsung TV also has a strong signature in the external data.

As a next step, we would take exactly the devices that are present in the Internal wireless medium and take their external data. Therefore, we would be performing clustering on the external data of the devices present only in the host network; and compare it to the results obtained from the external wireless clustering done previously to see if we get the same results.

Nest Thermostats had similar results. Initially they were in a cluster with other devices. When the clusters were 19, all the Nest thermostats went into an exclusive cluster. This trend continued and when the clusters were 31, they broke into two exclusive clusters with 3 Nest thermostats each (the same devices in each cluster), just like it happened in the external wireless clustering. They remain that way for some iterations and then increasingly break into more exclusive clusters.

Egauge devices, just like the previous External clustering, did not have a discernible signature. Up until the last iteration, they were present in a cluster with 15+ other devices. Figure 7.7 on page 116 shows how Egauge stays till the end, in a cluster where only 20% to 50% of the devices are Egauge.

Washing machine and Dryer were initially in a cluster with 10 other devices and gradually they broke into a cluster with fewer devices. When clusters were 26, they both went into an exclusive cluster of their own and until the last iteration they remain in this exclusive cluster. Hence, the washing machine and dryer also exhibit the same behaviour as they did in the previous external clustering.

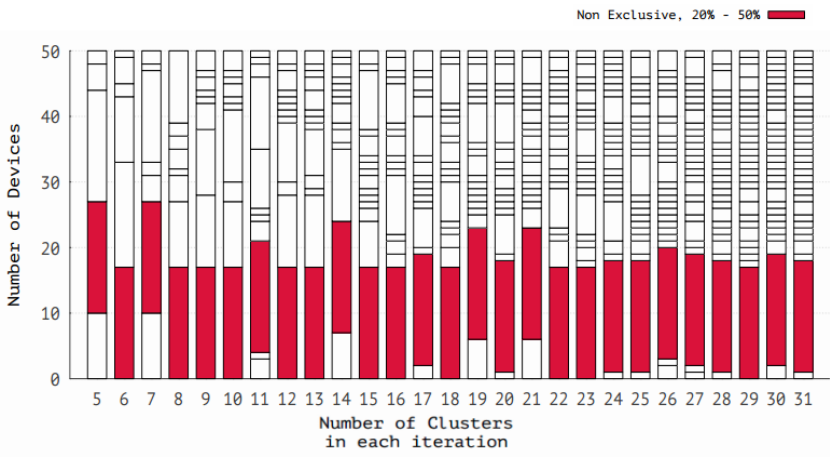

Figure 7.7: Clustering Progression of Egauges - External Wireless medium

The router, just like before, from the beginning till the end, remains in a cluster with other 15+ devices. But the Access Point, that is seen throughout the external data analysis, goes into its own exclusive cluster in the first iteration itself, when the clusters are 5 and remains that way till the last iteration.

Thus externally, irrespective of whether the clustering is done including all the devices observable at that point (292 devices) or considering only devices present at the host network (69), the behaviour exhibited by the devices remains largely unchanged. Their signatures and how quickly they converge does not change in either case. Hence, just by looking at the data available externally, without even joining the network, it is possible to group IoT devices together via clustering.

7.7 External vs. Internal clustering

Up until this point, we were analysing the clustering results of each section of data. Now we would compare the results of External and Internal clustering to find out their differences. For this comparison, we would take devices belonging to the host network alone.

Nest thermostats form exclusive clusters when the number of clusters are 19 in External Wireless and remain in this exclusive cluster until the number of clusters are 30. But in Internal Wireless, the Nest thermostats begin to break into different clusters (exclusively or non-exclusively) before becoming an exclusive cluster themselves. This seems like Nests have a stronger signature on the External wireless. But, we are considering a lot of parameters like protocols, ports etc., internally, that are more characteristic of the devices and considering only a few parameters externally. Therefore the data is more complex in the Internal wireless. Hence the internal goes a step further and distinguishes among the Nest devices.

Egauge devices have no discernible signature in the external wireless. In the internal wireless, their signature is better. They increasingly move into exclusive clusters as the iterations proceed. While for the Nest the external data was sufficient to produce a signature and the internal was more complex to distinguish even among the Nest devices, for the Egauges, the external data was not sufficient to produce a signature and the complex internal data was able to provide them strong signatures.

Washing machine and Dryer have strong signature both in the External and Internal wireless data. While both remained in an exclusive cluster till the end in external, they went into two exclusive clusters in Internal. This again proves that there was enough data to differentiate among the two internally.

Router had a very strong signal internally. It moved into an exclusive cluster when the number of clusters were 6 and remained so till the end. Externally, it did not have a discernible signature. It remained in a cluster with other 15+ devices till the end. This is probably because not all data from the router was captured externally due to insufficient range. The AP (ea:b3:5e) on the other hand, has very strong signature both internally and externally. It moved into exclusive cluster very early in both the clustering. As enough data from it was captured, it exhibited the same behaviour internally and externally. Samsung TV also had strong signatures both internally and

externally. It went into an exclusive cluster of its own in both clustering. In internal clustering, this happened earlier than external clustering, by 5 iterations.

By using the formula that we described earlier, we came with a quantitative value for the network signature and were able to compare the external and internal network signature strengths of various devices. Table 7.1 on page 118 shows the network signatures of various devices calculated both internally and externally. By comparing them, we can find out whether a device's internal signature is stronger than its external signature or vice versa.

Table 7.1: Network Signature of Devices

Device	**External**			**Internal**		
	Exclusive	Single / Multiple	Signature Strength	Exclusive	Single / Multiple	Signature Strength
Nest Thermostats	16	Single	0.516	8	Single	0.163
Washer and Dryer	12	Single	0.387	33	Single	0.674
Samsung TV	16	Single	0.516	14	Single	0.286
Raspberry Pi	Never	-		6	Single	0.122
Nintendo	8	Single	0.258	33, 38	Multiple	1.449
Apple Laptops	18, 9, 31	Multiple	1.871	15	Single	0.306
Egauge	Never	-		22, 30, 31	Multiple	1.694

Therefore, when we compare the internal and external clustering, we can see that many devices go into exclusive clusters early in the internal clustering compared to the external clustering. Because internal has very rich data that differentiates between the different types of devices easily. For some devices, they did not have discernible external signature but devices like Nest thermostats, Washing machine, Dryer, Samsung TV etc., have enough external data to give them strong signatures externally itself. Internally, since there is much more information like protocol and ports, which very strongly reflect the type of activity done by each device, internal clustering goes one step further and differentiates among the different devices of same type (Eg: Nest Thermostats) or if there is just one device, it goes into an exclusive cluster of its own much early.

7.8 Conclusion

Thus, we have taken the data that is available both internally and externally and fed them into clustering algorithm. We analysed the results and came up with a way of calculating the network signature of various devices to qualitatively visualize the signature strength of various devices. By comparing this signature, we were able to say that in many cases, the network signature of a device was stronger on the internal data when compared to the external data. But we also found that the external data had enough information to give a discernible network signature for many devices albeit weaker than the internal. From this external signature, we were able to identify many devices like Nest or the Washing machine proving that the external data that is leaked outside holds information that is good enough to identify devices.

Also. by using the sum of squared differences between centroids, we were able to identify top parameters that influenced the clusters to separate in a particular manner, at every step of clustering. Those parameters that influenced a device towards a particular cluster or direction would be the parameters that are characteristic of the device and constitute the profile of the device and helps in identifying the devices from the network data. By using this information, we can also profile a smart home. From our data, we found that the internally collected data holds rich information to profile a smart home by identifying the various types of devices present. But externally collected data does not hold such a rich information to aid in a clear profiling of our data collection site.

CHAPTER 8

FUTURE WORK

As a first step in understanding IoT devices, we collected real-time data for a maximum period of 6 months from both outside and inside the network. We analysed the collected data and identified parameters on both sides that differentiated IoT and non-IoT devices. This showed us, in general, how IoT devices behaved differently in the network when compared to non-IoT devices. From the data, especially internally collected data, we were able to identify device specific markers for a lot of devices. Then we performed clustering on both the internal and external data to group similar devices and learn their network signature. We also came up with a way of quantifying the network signature of the devices to qualitatively visualize their signature strength. Then we investigated the difference in the signature strength between the internal and external data of the various types of devices and established that, though the internal signature was stronger than the external signature for many devices, the external data holds enough information to discern network signatures for at least some of the devices. We also made an effort at profiling the smart home with this data and found that, at our site, the internal data enabled a clear profiling of the site revealing the various types of devices present. But the external data was not good enough for this task.

As a next step, this experiment could be carried out in another two or three real-time locations and the results of the analysis from all locations can be compared to see if particular type of devices behave the same way on all networks. This would enable in creating a more generalized profile for each type of device that can be

applied to the network data from any network to identify those devices. Our way of calculating the network signature strength can be expanded with the help of data that is collected from all the real-time locations and the formula can be standardized to be used for any device from any network, stepping it up from its current usage for qualitative measurement. The internal data has not been completely analyzed for our experiment. The rich internal data can be analyzed more deeply to reveal even more device specific characteristics of various devices. This could also help in enriching the profile of the devices. All the above would help in understanding IoT devices better and would help in defining more robust and tailor-made security solutions for network connected IoT devices.

BIBLIOGRAPHY

[1] D. Evans, "The internet of things: How the next evolution of the internet is changing everything. cisco," *Int. J. Internet*, vol. 3, no. 2, pp. 123–132, 2011.

[2] K. Ashton *et al.*, "That internet of things thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[3] D. Pangburn, "Even this data guru is creeped out by what anonymous location data reveals about us," *URL https://www.fastcompany.com/3068846/how-your-location-data-identifies-you-gilad-lotan-privacy*, 2017.

[4] A. Rutkin, "Speech recognition ai identifies you by voice wherever you are," *URL https://www.newscientist.com/article/mg22830423-100-speech-recognition-ai-identifies-you-by-voice-wherever-you-are/*, 2015.

[5] S. Cobb, "things to know about the october 21 iot ddos attacks,," *WeLiveSecurity, In press release*, vol. 24, 10.

[6] B. Krebs, "Krebsonsecurity hit with record ddos.(2016)," *URL https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos*, 2016.

[7] H. Kagermann, W. Wahlster, and J. Helbig, "Securing the future of german manufacturing industry," *Recommendations for implementing the strategic initiative INDUSTRIE*, vol. 4, 2013.

[8] A. Greenberg, "Hackers remotely kill a jeep on the highwaywith me in it," *Wired*, vol. 7, p. 21, 2015.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[10] M. Strohbach, H. Ziekow, V. Gazis, and N. Akiva, "Towards a big data analytics framework for iot and smart city applications," in *Modeling and processing for next-generation big-data technologies*, pp. 257–282, Springer, 2015.

[11] S. M. Oteafy and H. S. Hassanein, "Iot in the fog: A roadmap for data-centric iot development," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 157–163, 2018.

[12] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: a review," in *Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on*, vol. 3, pp. 648–651, IEEE, 2012.

[13] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, p. 5, ACM, 2015.

[14] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[15] D. Bekerman, B. Shapira, L. Rokach, and A. Bar, "Unknown malware detection using network traffic classification," in *Communications and Network Security (CNS), 2015 IEEE Conference on*, pp. 134–142, IEEE, 2015.

[16] A. Sivanathan, D. Sherratt, H. H. Gharakheili, V. Sivaraman, and A. Vishwanath, "Low-cost flow-based security solutions for smart-home iot devices," in *Advanced Networks and Telecommunications Systems (ANTS), 2016 IEEE International Conference on*, pp. 1–6, IEEE, 2016.

[17] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "Profiliot: a machine learning approach for iot device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, pp. 506–509, ACM, 2017.

[18] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[19] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized iot devices using machine learning techniques," *arXiv preprint arXiv:1709.04647*, 2017.

[20] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and classifying iot traffic in smart cities and campuses," in *Computer Communications Workshops (INFOCOM WKSHPS), 2017 IEEE Conference on*, pp. 559–564, IEEE, 2017.

[21] G. Combs, "Tsharkdump and analyze network traffic," *Wireshark*, 2012.

[22] G. Combs *et al.*, "Wireshark-network protocol analyzer," *Version 0.99*, vol. 5, 2008.